iSeries

CL Commands Volume 21

iSeries

CL Commands Volume 21

Contents

Command Descriptions 1
SBMCRQ (Submit Change Request) Command Description 1
SBMDBJOB (Submit Database Jobs) Command Description 2
SBMDKTJOB (Submit Diskette Jobs) Command Description 5
SBMFNCJOB (Submit Finance Job) Command Description 8
SBMJOB (Submit Job) Command Description 10
SBMJOBJS (Submit Job using Job Scheduler) Command Description 21
SBMNETJOB (Submit Network Job) Command Description 33
SBMNWSCMD (Submit Network Server Command) Command Description 35
SBMRMTCMD (Submit Remote Command) Command Description 38
TRCASPBAL (Trace ASP Balance) Command Description 42
TRCCNN (Trace Connection) Command Description 44
TRCCPIC (Trace CPI Communications) Command Description. 50
TRCICF (Trace ICF) Command Description 52
TRCINT (Trace Internal) Command Description 55
TRCJOB (Trace Job) Command Description 70
TRCREX (Trace REXX) Command Description 75
TRACEROUTE Command Description. 77
TRCTCPAPP (Trace TCP/IP Application) Command Description 77
TRCTCPRTE (Trace TCP/IP Route) Command Description 87
TFRBCHJOB (Transfer Batch Job) Command Description 91
TFRCTL (Transfer Control) Command Description 93
TFRJOB (Transfer Job) Command Description. 95
TFRPASTHR (Transfer Pass-Through) Command Description 98
TFRSECJOB (Transfer Secondary Job) Command Description. 99
TFRGRPJOB (Transfer to Group Job) Command Description 99
UPDPGM (Update Program) Command Description 102
UPDSRVPGM (Update Service Program) Command Description 107
UPDSYSINF (Update System Information) Command Description 113
VRYCFG (Vary Configuration) Command Description 114
APING (Verify APPC Connection) Command Description 121
VFYAPPCCNN (Verify APPC Connection) Command Description 122
VFYCMN (Verify Communications) Command Description 124
VFYIMGCLG (Verify Image Catalog) Command Description 126
VFYLNKLPDA (Verify Link Supporting LPDA-2) Command Description 127
VFYMOVBRM (Verify Moves Using BRM) Command Description 131
VFYNTWAUTE (Verify NetWare Authentication Entry) Command Description 132
VFYOPT (Verify Optical) Command Description 133
VFYOPCCNN (Verify OptiConnect Connections) Command Description 133
VFYPRT (Verify Printer) Command Description 134
VFYTAP (Verify Tape) Command Description 135
VFYTCPCNN (Verify TCP/IP Connection) Command Description 136
WAIT (Wait) Command Description 139
WRKACTJOB (Work with Active Jobs) Command Description. 141
WRKALRD (Work with Alert Descriptions) Command Description 143
WRKALRTBL (Work with Alert Tables) Command Description 145
WRKALR (Work with Alerts) Command Description 147
WRKAPPNSTS (Work with APPN Status) Command Description 150
WRKASPBRM (Work with ASP Descriptions) Command Description 152
WRKAUT (Work with Authority) Command Description 153
WRKAUTL (Work with Authorization Lists) Command Description 155
WRKBNDDIR (Work with Binding Directories) Command Description 156
WRKBNDDIRE (Work with Binding Directory Entries) Command Description 158

WRKCALBRM (Work with Calendars using BRM) Command Description 159
WRKCRQD (Work with Change Request Descriptions) Command Description. 160
WRKCHTFMT (Work with Chart Formats) Command Description 161
WRKCOSD (Work with Class-of-Service Descriptions) Command Description 163
WRKCLS (Work with Classes) Command Description. 163
WRKCLSBRM (Work with Classes using BRM) Command Description 165
WRKCMD (Work with Commands) Command Description 166
WRKCMTDFN (Work with Commitment Definitions) Command Description 168
WRKCSI (Work with Communications Side Information) Command Description 171
WRKCFGL (Work with Configuration Lists) Command Description 173
WRKCFGSTS (Work with Configuration Status) Command Description 174
WRKCNNLE (Work with Connection List Entries) Command Description. 177
WRKCNNL (Work with Connection Lists) Command Description. 178
WRKCNTINF (Work with Contact Information) Command Description 179
WRKCNRBRM (Work with Containers using BRM) Command Description 180
WRKCTLGBRM (Work with Control Groups using BRM) Command Description 180
WRKCTLD (Work with Controller Descriptions) Command Description 181
WRKDTAARA (Work with Data Areas) Command Description. 182
WRKDTADFN (Work with Data Definitions) Command Description 183
WRKDTADCT (Work with Data Dictionaries) Command Description 184
WRKDTAQ (Work with Data Queues) Command Description 185
WRKDBFIDD (Work with Database Files Using IDDU) Command Description 186
WRKDEVD (Work with Device Descriptions) Command Description 187

Command Descriptions

»

SBMCRQ (Submit Change Request) Command Description

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

SBMCRQ Command syntax diagram

Purpose

The Submit Change Request (SBMCRQ) command submits a change request to be run.

Restriction: You must have *USE authority to the change request description object that the submitted change request is based on.

The following notes provide information about how the command works.

Notes:

1. If the USRPRF attribute of the change request description specified is *SBM, this submitted change request runs under the user profile that is running the SBMCRQ command. If the USRPRF attribute of the change request description is *OWNER, this submitted change request runs under the user profile of the owner of the change request description. For more information on the user profile attribute, see the USRPRF parameter of the Create Change Request Description (CRTCRQD) command and of the Change Request Description (CHGCRQD) command.
2. If a node list object is referred in the change request and the node list changes after the change request submission, the submitted change request is not affected. If the change request description of the submitted change request is deleted after the submission is complete, the submitted change request is not affected.
3. If System Manager licensed program is not active, an information message is sent to the user of the SBMCRQ command. The message indicates that the submitted change request is not processed until the System Manager licensed program starts.

Required Parameter

CRQD Specifies the qualified name of the change request description object.

The possible library values are:

<u>*LIBL:</u> Locates the change request description in the library list of the current job.

***CURLIB:** Locates the change request description in the current library. If no library is specified as the job’s current library, the QGPL library is used.

library-name: Search only the library specified in this parameter.

change-request-description-name: Specify the name of the change request description on which the change request to submit is based.

Optional Parameter

CRQ Specifies the change request name and the sequence number of the submitted change request. The sequence number is a 6-digit number that uniquely defines the change request.

Element 1: Change Request Name

***CRQD:** Use the same name as the change request description specified.

change-request-name: Specify the name of the change request to submit.

Element 2: Sequence Number

***GEN:** Generates a sequence number. The initial sequence number is 000010. Subsequent sequence numbers are generated in increments of ten. If the sequence number generated is already in use, the next increment of ten is used.

sequence-number: Specify the sequence number. The valid values range from 1 to 999999.

Examples for SBMCRQ

Example 1: Submitting a Change Request

```
SBMCRQ CRQD(MYLIB/CHG222) CRQ(*CRQD 123)
```

This command submits a change request based on the change request description CHG222 in MYLIB. The change request name submitted is CHG222 with a sequence number of 123.

Example 2: Submitting a Change Request

```
SBMCRQ CRQD(*CURLIB/CHG999) CRQ(CRQ0001 *GEN)
```

This command shows how to submit change request CRQ0001 with sequence number 00010 if it is not already being used. The submitted change request is based on change request description CHG999 in the current library.

Error messages for SBMCRQ

*ESCAPE Messages

None «

SBMDBJOB (Submit Database Jobs) Command Description

SBMDBJOB Command syntax diagram

Purpose

The Submit Database Jobs (SBMDBJOB) command allows a job that is running to submit other jobs to job queues to be run as batch jobs. The input stream is read either from a physical database file or from a logical database file in single-record format. This command specifies the name of the database file and member from which the jobs are read, the name of the job queue to be used if the Batch Job (BCHJOB) command has JOBQ(*RDR) specified, and whether jobs being submitted can be displayed by the Work with Submitted Jobs (WRKSBMJOB) command.

A Submit Database Jobs operation reads the file once and ends when the end-of-file is read or when an End Input (ENDINP) command (a delimiter) is encountered. The ENDINP command (a delimiter) is not recognized if it is within an inline file that ends with characters that are not default ending characters (as specified in the ENDCHAR parameter of the Data (DATA) command). The SBMDBJOB operation can be canceled either by canceling the request from the system request menu or by canceling the job in which the process is running.

In contrast to a spool reader started with the Start Database Reader (STRDBRDR) command, the SBMDBJOB command operates in the same process as the requesting function and does not do syntax checking on the input stream.

Restriction: The specified database file either must consist of single-field records and must have an arrival sequence access path, or it must be a standard database source file.

Required Parameter

FILE Specifies the name of the database file from which the input stream is read.

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 the database file.

Optional Parameters

MBR Specifies the name of the member in the specified file that contains the input stream being read.

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

file-member-name: Specify the name of the member that contains the input stream being read.

JOBQ Specifies the job queue on which the job entries are placed. A job entry is placed on this queue for each job in the input stream that has JOBQ(*RDR) specified on the Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified on the BCHJOB command or in the job description is used. (The job queue for each job in the input stream can be different.) This parameter is valid only if ACTION(*SUBMIT) is specified on this command, in the existing network job entry, or in a subsequent Change Network Job Entry (CHGNETJOBE) command.

Note: If both the user identified in the job description of the job being read and the user processing the Submit Database Job (SBMDBJOB) command are not authorized to the job queue on which the job should be placed, the job ends and a diagnostic message is placed in the job log. The input stream, continues to be processed, starting with the next job. If either user is authorized to the job queue, the job runs without error.

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

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

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

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

QBATCH: The job entry is placed on the QBATCH job queue, which is the default job queue, if JOBQ(*RDR) is specified on the BCHJOB command.

job-queue-name: Specify the name of the job queue to which each job in the job stream is sent if JOBQ(*RDR) is specified on the BCHJOB command.

DSPSBMJOB

Specifies whether the jobs being submitted can be displayed on the submitted jobs display. Any submitted job of the type specified by the SBMFROM parameter of the WRKSBMJOB command can be displayed if *YES is specified on this parameter.

***YES:** This job can be displayed by the WRKSBMJOB command.

***NO:** This job is not displayed on any display produced by the WRKSBMJOB command.

Example for SBMDBJOB

```
SBMDBJOB  FILE(QGPL/BILLING)
```

This command submits jobs using input from the database file named BILLING, which is in the QGPL library. The first member in the BILLING file contains the input stream to be processed. The default system-supplied job queue QBATCH is used.

Error messages for SBMDBJOB

***ESCAPE Messages**

CPF1751

Error while processing job &3/&2/&1.

CPF1754

File &1 in library &2 not database file or DDM file.

CPF1760

Submit jobs command not allowed.

CPF2207

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

CPF3307

Job queue &1 in &2 not found.

CPF3330

Necessary resource not available.

CPF3363

Message queue &1 in library &2 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9812

File &1 in library &2 not found.

CPF9815

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

CPF9845
Error occurred while opening file &1.

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

SBMDKTJOB (Submit Diskette Jobs) Command Description

SBMDKTJOB Command syntax diagram

Purpose

The Submit Diskette Jobs (SBMDKTJOB) command allows jobs that are running to submit other jobs to job queues to be run as batch jobs. The input stream is read from a diskette. This command specifies:

- The name of the diskette device
- The label identifier
- The volume, creation date, and interchange code type of the file containing the input stream
- The names of the job queue and message queue used
- Whether jobs being submitted can be displayed by the Work with Submitted Jobs (WRKSBMJOB) command

A Submit Diskette Jobs operation reads the file once and ends when the end-of-file is read or when an ENDINP (End Input) command (delimiter) is encountered. The ENDINP command (delimiter) is not recognized if it is within an inline file that ends with characters that are not default ending characters (as specified in the ENDCHAR parameter of the DATA command). The SBMDKTJOB operation can be canceled either by canceling the request from the system request menu or by canceling the job in which the operation is running.

In contrast to a spool reader started with the Start Diskette Reader (STRDKTRDR) command, the SBMDKTJOB command operates in the same process as the requesting function and does not do syntax checking on the input stream.

Note: This command cannot be used to read the data files of diskettes that are in the save/restore format.

Required Parameters

DEV Specifies the name of the diskette device used to read the input stream. Specify the name of the diskette device.

LABEL
Specifies the data file label (identifier) of the diskette data file on the diskette that contains the input stream. Specify the data file identifier. The data file identifier can be no longer than eight characters.

Optional Parameters

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

***NONE:** No volume identifiers are specified. The current volume is used.

volume-identifier: Specify the identifiers of one or more volumes in the order that they are put on the diskette device and used. Each volume identifier contains up to six characters. A blank is used as the separator character when listing multiple identifiers.

CRTDATE
Specifies the creation date of the object.

Note: The creation date should not be specified if the date is not checked. If the date written on the diskette containing the data file does not match the date specified here, an error message is sent to the message queue named in the MSGQ parameter.

***NONE:** The creation date is not specified, and no check is made.

creation-date: Specify the creation date of the data file to be read. The date should be in the format specified by the QDATFMT system value.

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 code is used.

JOBQ Specifies the job queue on which the job entries are placed. A job entry is placed on this queue for each job in the input stream that has JOBQ(*RDR) specified on the Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified on the BCHJOB command or in the job description is used. (The job queue for each job in the input stream can be different.) This parameter is valid only if ACTION(*SUBMIT) is specified on this command, in the existing network job entry, or in a subsequent Change Network Job Entry (CHGNETJOBE) command.

Note: If both the user identified in the job description of the job being read and the user processing the Submit Diskette Job (SBMDKTJOB) command are not authorized to the job queue on which the job should be placed, the job ends and a diagnostic message is placed in the job log. The input stream, continues to be processed, starting with the next job. If either user is authorized to the job queue, the job runs without error.

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

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

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

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

QBATCH: The job entry is placed on the QBATCH job queue, which is the default job queue if JOBQ(*RDR) is specified on the BCHJOB command.

job-queue-name: Specify the name of the job queue to which each job in the job stream is sent if JOBQ(*RDR) is specified on its BCHJOB command.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***DEVD:** The messages are sent to the message queue indicated by the device description of the device being used.

***REQUESTER:** The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, *OUTQ is used.

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

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

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

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

message-queue-name: Specify the name of the message queue to which operational messages are being sent.

DSPSBMJOB

Specifies whether the jobs being submitted can be displayed on the submitted jobs display. Any submitted job of the type specified by the SBMFROM parameter of the WRKSBMJOB command can be displayed if *YES is specified.

***YES:** This job can be displayed by the WRKSBMJOB command.

***NO:** This job is not displayed on any display produced by the WRKSBMJOB command.

Example for SBMDKTJOB

```
SBMDKTJOB  DEV(QDKT)  LABEL(OCT24)  VOL(SALES)
```

This command submits diskette jobs using diskette input from the device QDKT. The submit diskette jobs function gets its input from the data file named OCT24 with the volume identifiers SALES. The default job queue QBATCH is used as the receiving job queue when JOBQ(*RDR) is found in the job description. Operational messages are sent to the message queue defined by the device.

Error messages for SBMDKTJOB

***ESCAPE Messages**

CPF1751

Error while processing job &3/&2/&1.

CPF1752

Device &1 not correct device type.

CPF1760

Submit jobs command not allowed.

CPF2207

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

CPF3307

Job queue &1 in &2 not found.

CPF3330

Necessary resource not available.

CPF3363
Message queue &1 in library &2 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9814
Device &1 not found.

CPF9845
Error occurred while opening file &1.

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

***STATUS Messages**

CPF1755
Reading job &3/&2/&1 from volume &4.

SBMFNCJOB (Submit Finance Job) Command Description

SBMFNCJOB Command syntax diagram

Purpose

The Submit Finance Job (SBMFNCJOB) command submits a batch job so the user's finance application programs communicates with the 4701 or 4702 control unit application programs.

Use the SBMFNCJOB command only if:

- Communicating with a 4701 or 4702 control unit
- A device table and a program table have been defined using the Work with Device Table (WRKDEVTBL) and Work with Program Table (WRKPGMTBL) commands; defining a user table using the Work with User Table (WRKUSRTBL) command is optional
- The user's 4701 or 4702 control unit application program sends data (transactions) first and expects a response
- The user's 4701 or 4702 control unit application program passes data in the proper format

Restriction: This command is shipped with public *EXCLUDE authority.

Required Parameters

DEVTBL
Specifies the name of the device table that the finance job uses to determine which 4704 or 3624 devices it controls.

PGMTBL
Specifies the name of the program table that the finance job uses to determine, from the program ID (sent in the data stream with a finance transaction), which system user program names will process the finance transaction.

Optional Parameters

USRTBL
Specifies the name of the user table that the finance job uses to verify a valid finance user when a finance sign-on is received.

***NONE:** No user IDs are verified.

user-table-name: Specify the name of a user table that defines user IDs for the 4700 device.

JOB Specifies the job name that is associated with the submitted finance job. More information on this parameter is in Commonly used parameters.

QFNCJOB: The job name is submitted as QFNCJOB.

job-name: Specify the job name that is associated with the submitted finance job.

JOBD Specifies the qualified name of the job description that is used by the finance job.

The name of the finance job 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.

QFNC: The submitted finance job uses the job description QFNC.

job-description-name: Specify the name of a job description that is used by the finance job. (If no library name is given, the job description is found through the library list used by the job in which the SBMFNCJOB command is entered.)

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***WRKSTN:** The finance messages are sent to the message queue of the work station from which the finance job was submitted.

***NONE:** No finance messages are sent to a message queue.

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

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

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

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

message-queue-name: Specify the name of the message queue to which messages are sent. (If no library name is given, the library list of the job issuing the SBMFNCJOB command is used to find the queue.)

Examples for SBMFNCJOB

Example 1: Submitting a Batch Job that Communicates with Devices it Acquires

```
SBMFNCJOB  DEVTBL(DEVTBL1)  PGMTBL(PGMTBL1)
  USRTBL(USRTBL1)
```

This command submits batch job QFNCJOB. The job communicates with all devices it acquires from device table DEVTBL1, allowing users whose user IDs are found in USRTBL1 to sign on the devices. Each transaction sent by the finance devices is processed by determining, in PGMTBL1, which application program must be called, then it calls that program.

The job description used by the finance job in this example is QFNC.*LIBL. Messages sent as a result of the finance job are sent to the message queue of the work station from which the job was submitted.

Example 2: User IDs Not Verified

```
SBMFNCJOB  DEVTBL(DEVTBL2)  PGMTBL(PGMTBL2)
  JOB(CTFJOB)  JOBD(CTFJOBD) MSGQ(*NONE)
```

This command submits batch job CTFJOB. CTFJOB runs under job description CTFJOBD and does not send messages to any work station message queue while running. No verification of user IDs is performed by the finance job.

Error messages for SBMFNCJOB

***ESCAPE Messages**

CPF8382

Finance job cannot be processed.

SBMJOB (Submit Job) Command Description

SBMJOB Command syntax diagram

Purpose

The Submit Job (SBMJOB) command allows a job that is being run to submit another job to a job queue to be run later as a batch job. Only one element of request data can be placed in the new job’s message queue. The request data can be a CL command if the routing entry used for the job specifies a CL command processing program.

Note: A job started by the SBMJOB command uses the accounting code of the job that submits the job. The accounting code specifications on the submitted jobs’ JOBD and USRPRF parameters are ignored.

Restrictions:

This command is conditionally threadsafe. The following restrictions apply:

1. » To submit a job that runs under a different user profile, you must have *USE authority to that user profile.
2. The user that issues the Submit Job command must have
   - *USE authority to the command specified by the CMD (command to run) parameter and *EXECUTE authority to the library containing that command.
   - *READ authority to the job description (JOBD) and *EXECUTE authority to the library containing that job description.
   - *USE authority to the job queue (JOBQ) and *EXECUTE authority to the library containing that job queue.
   - *USE plus *ADD authority to the message queue (MSGQ) and *EXECUTE authority to the library containing that message queue.

- *USE authority to the sort sequence table (SRTSEQ) and *EXECUTE authority to the library containing that sort sequence table.
- *EXECUTE authority to all auxiliary storage pool (ASP) device descriptions in the initial ASP group (INLASPGRP).

3. The user for the submitted job must have
   - *USE authority to the job description (JOBD).
   - *READ authority to the output queue (OUTQ) and *EXECUTE authority to the library containing that output queue.
   - *USE authority to all auxiliary storage pool (ASP) device descriptions in the initial ASP group (INLASPGRP).
   - *USE authority to the library specified for the current library (CURLIB) parameter.
   - *USE authority to all the libraries specified for the initial library list (INLLIBL) parameter. «
4. If a Job Notification Exit Point has been registered to send a message to a DDM data queue whenever a Submit Job is done, the message will not be sent if the SBMJOB command is issued in a multithreaded job. For more information on the Job Notification function, refer to the Job Notification Exit Point in the Work Management chapter of the Application Program Interfaces (APIs) topic in the Information Center.

Optional Parameters

JOB Specifies the job name that is associated with the submitted job while it is being processed by the system.

***JOBD:** The simple name of the job description used with this job is the name of the job itself.

job-name: Specify the simple name of the job that is used while it is being processed by the system.

JOBD Specifies the job description used to submit jobs for batch processing.

***USRPRF:** The job description specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

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.

USER Specifies the name of the user profile under which the job is submitted. USER(*JOBD) is not valid when USER(*RQD) is specified on the Create Job Description (CRTJOBD) command.

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

***JOBD:** The user profile name in the specified job description is used for the job being submitted.

When the system is running under security level 40, the user of this command must be authorized to the user specified in the job description.

user-name: Specify the user profile name that is used for the submitted job. The user must be authorized to the user profile and the job description; the user profile must also be authorized to the job description (JOBD).

JOBQ Specifies the qualified name of the job queue on which this job is placed.

***JOBD:** The submitted job is placed on the job queue named in the specified job description.

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

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

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

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

job-queue-name: Specify the qualified name of the job queue on which the submitted job is placed.

JOBPTY

Specifies the scheduling priority for the submitted job. Valid values range from 1 through 9, where 1 is the highest priority and 9 is the lowest priority. More information on this parameter is in Commonly used parameters.

***JOBD:** The scheduling priority specified in the job description is used.

scheduling-priority: Specify a value, ranging from 1 through 9, for the scheduling priority for the job.

OUTPTY

Specifies the output priority for spooled files that are produced by the submitted job. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in Commonly used parameters.

***JOBD:** The output priority specified in the job description is used for the job.

output-priority: Specify a value, ranging from 1 through 9, for the priority of the output files of the submitted job.

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.

***CURRENT:** The same print text of the submitting job is used.

***JOBD:** The value in the job description is used.

***SYSVAL:** The system value (QPRTTXT) is used.

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

'print-text': Specify the character string that is printed at the bottom of each page. Up to 30 characters can be entered, enclosed in apostrophes.

RTGDTA

Specifies the routing data that is used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program in which the job runs.

QCMDB: The routing data used by the IBM-supplied batch subsystem, QBATCH, to route batch jobs to the IBM-supplied control language processor QCMD is used.

***JOBD:** The routing data specified in the job description is used to start the routing step.

***RQSDTA:** Up to 80 characters of the request data specified in the RQSDTA parameter of this command is used as the routing data for the job.

'routing-data': Specify the character string that is used as the routing data for starting the job. Up to 80 characters can be entered enclosed in apostrophes, if necessary.

CMD Specifies the command that runs in the submitted job. The IBM-supplied default routing program QCMD must be used when the job is started or the job will not run. Because the command you specify is used for the request data, the value specified on the RQSDTA parameter in the job description is ignored. The command can be a maximum of 20000 characters in length.

RQSDTA

Specifies the request data that is placed in the submitted job's message queue. For example, if RTGDTA (QCMDB) is specified, the IBM-supplied batch subsystem, QBATCH, is used, and a CL command is supplied, it becomes a message that is read by the control language processor, QCMD.

***CMD:** The input from the CMD parameter is placed in this job's message queue.

***JOBD:** The request data specified in the job description used by the job is placed in this job's message queue.

***NONE:** No request data is placed in the job's message queue.

***RTGDTA:** The routing data specified in the RTGDTA parameter of this command is placed as the last entry in the job's message queue.

'request-data': Specify the character string that is placed as the last entry in the submitted job's message queue. Up to 3000 characters can be entered, enclosed in apostrophes, if necessary. When a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used *inside* the command, double apostrophes must be used instead.

SYSLIBL

Specifies the system portion of the library being used by the submitted job.

***CURRENT:** The system portion of the library list of the submitting job is used.

***SYSVAL:** The library list specified in the system value (QSYSLIBL) at the time the job is started is used for the submitted job.

CURLIB

Specifies the name of the library being used as the current library for jobs initiated by this user profile.

***CURRENT:** The current library for the job is used for the submitted job.

***USRPRF:** The current library specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***CRTDFT:** There is no current library for the submitted job. If objects are created in the current library, the QGPL library is used as the default current library.

current-library-name: Specify the name of the library that is used as the current library of the submitted job.

INLLIBL

Specifies the user portion of the first library list that is used by the submitted job to search for any object names that are specified without a library qualifier. This does not include the system portion of the library list.

Note: Duplication of libraries in the library list is not allowed.

***CURRENT:** The user portion of the library list being used by the current job is used for the submitted job.

***JOBD:** The library list specified in the job description used with the job is used as the first library list for the job.

***SYSVAL:** The system default library list, QUSRLIBL, is used by the job.

***NONE:** The user portion of the first library list is empty; only the system portion is used.

library-name: Specify the names of one or more libraries that are in the user portion of the library list and are used by the submitted job. No more than 250 names can be specified. The libraries are searched in the same order as they are listed.

LOG Specifies the message logging values used to determine the amount and type of information sent to the job log by this job. This parameter has three elements: the message (or logging) level, the message severity, and the level of message text. If no values are specified on this parameter, the values specified in the job description associated with this job are used.

Element 1: Message Level

***JOBD:** The value specified for message logging in the job description is used for the submitted job.

message-level: Specify a value, ranging from 0 through 4, that specifies the message logging level used for the submitted job's messages. For additional information on the message levels, refer to "Message Level" under the LOG parameter of the Create Job Description (CRTJOBD) command.

Element 2: Message Severity

***JOBD:** The value specified for message logging in the job description is used for the submitted job.

message-severity: Specify a value, ranging from 00 through 99, that is used in conjunction with the logging level to determine which error messages are logged in the job log. More information on this parameter is in Commonly used parameters.

Element 3: Message Text Level

***JOBD:** The value specified for message logging in the job description is used for the submitted job.

***MSG:** Only the message text is written to the job log.

***SECLVL:** Both the message text and the message help (cause and recovery) of the error message are written to the job log.

***NOLIST:** If the job ends normally, no job log is produced. If the job ends abnormally (if the job end code is 20 or higher), a job log is produced. The messages that appear in the job log contain both the message text and the message help.

LOGCLPGM

Specifies whether the commands that have run in a control language program are logged to the job log by way of the CL program's message queue. This parameter sets the status of the job's logging flag. If *NO is specified, the logging flag status is off and CL commands are not logged. If *YES is specified and the LOG (*JOB) value is specified in the Create CL Program (CRTCLPGM) command, all commands in the CL program that can be logged are logged to the job log.

For more information on request logging, refer to the LOG parameter in the CRTCLPGM command description.

***JOBD:** The value specified in the job description is used.

***NO:** The commands in a CL program are not logged to the job log.

***YES:** Commands in a CL program are logged to the job log.

INQMSGRPY

Specifies the way that predefined messages that are sent as a result of running this job are answered. The user can specify that no change is made in the way that predefined messages are answered, that all inquiry messages require a reply, that a default reply be issued, or that the system reply list is checked for a matching reply as each predefined inquiry message is sent. Refer to the Add Reply List Entry (ADDRPYLE) command description for more information.

***JOBD:** The inquiry message reply control specified in the job description used with this job is used.

***RQD:** A reply is required by the receiver of the inquiry message for all inquiry messages that occur when this command is run.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

***SYSRPYL:** The system reply list is checked to see if there is an entry for an inquiry message that is issued as a result of running this job that has a message identifier and any comparison data that match the inquiry message identifier and message data. If a match occurs, the reply value in that entry is used. If no entry exists for that message, a reply is required.

PRTDEV

Specifies the name of the default printer device for this job. If the printer file being used to create the output specifies to spool the file, the spooled file is placed on the device's output queue, which is named the same as the device.

Note: This assumes the defaults are specified on the OUTQ parameter for the printer file, job description, user profile and workstation.

***CURRENT:** The printer device being used by the job that is currently running is used by the submitted job.

***USRPRF:** The printer device specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

***JOBD:** The printer device specified in the job description is used for the submitted job.

printer-device-name: Specify the name of the printer device that is used for the submitted job.

OUTQ Specifies the qualified name of the output queue used for spooled files that specify OUTQ(*JOB). This parameter applies only to printer files that have *JOB specified on the OUTQ parameter.

***CURRENT:** The output queue used by the current job is used for the submitted job.

***USRPRF:** The output queue specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***DEV:** The output queue specified on the PRTDEV parameter is used.

***JOBD:** The output queue named in the job description used with the submitted job is the job's default output queue.

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

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

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

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

output-queue-name: Specify the qualified name of the output queue that is used as the default output queue by the submitted job.

HOLD Specifies whether jobs using this job description are placed on the job queue in the hold condition. A job placed on the job queue in the hold condition is held until it is either released by the Release Job (RLSJOB) command or canceled by the End Job (ENDJOB) or Clear Job Queue (CLRJOBQ) command. If the job is not run before the next power-down of the system, the job queue can be cleared (and the job ended) when the next initial program load (IPL) is done.

***JOBD:** The value specified in the job description determines whether the job is held when it is put in the job queue.

***NO:** The job is not held when it is put in the job queue.

***YES:** The job is held when it is put on the job queue until it is released by a Release Job (RLSJOB) command or ended by a End Job (ENDJOB) command.

SCDDATE

Specifies the date on which the submitted job becomes eligible to run.

If your system or your job is configured to use the Julian date format, the *MONTHSTR and *MONTHEND values are calculated as if the system or job did not use the Julian date format.

***CURRENT:** The submitted job becomes eligible to run on the current date.

***MONTHSTR:** The submitted job becomes eligible to run on the first day of the month. If you specify *MONTHSTR, and if today is the first day of the month, and if the time you specify on the SCDTIME parameter has not passed, the job is eligible to run today. Otherwise, the job becomes eligible to run on the first day of the next month.

***MONTHEND:** The submitted job becomes eligible to run on the last day of the month. If you specify *MONTHEND, and if today is the last day of the month, and if the time you specify on the SCDTIME parameter has not passed, the job is eligible to run today. Otherwise, the job becomes eligible to run on the last day of the next month.

***MON:** The job becomes eligible to run on Monday.

***TUE:** The job becomes eligible to run on Tuesday.

***WED:** The job becomes eligible to run on Wednesday.

***THU:** The job becomes eligible to run on Thursday.

***FRI:** The job becomes eligible to run on Friday.

***SAT:** The job becomes eligible to run on Saturday.

***SUN:** The job becomes eligible to run on Sunday.

date: Specify a date in the job date format with or without separators.

SCDTIME
Specifies the time on the scheduled date at which the job becomes eligible to run.

Note: Although the time can be specified to the second, the activity involved in submitting a job and the load on the system may affect the exact time at which the job becomes eligible to run.

The order that job entries with identical SCDDATE and SCDTIME values appear on the job queue may be different than the order in which they arrived. Likewise, these jobs may leave the job queue to be processed in an order different than the order in which they were entered. Do not assume jobs are entered or processed sequentially when they are scheduled to start at exactly the same time.

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

time: Specify a time in the system format with or without separators defined for the job.

DATE Specifies the date that is assigned to the submitted job when it is started.

***JOBD:** The date specified in the job description is used as the job date.

***SYSVAL:** The value in the QDATE system value at the time the job is started is used as the job date.

job-date: Specify the value that is used as the job date when the job is started. The date must be in the job date format. Possible formats are in the CL Programming book.

SWS Specifies the first settings for a group of eight job switches that are used with the submitted job. These switches can be set or tested in a CL program and used to control the flow of the program. For example, if a certain switch is on, another program can be called. The job switches may also be valid in other high-level language programs. Only 0's (off) and 1's (on) can be specified in the 8-digit character string.

***JOBD:** The value specified in the job description is the first setting for the job's switches.

switch-settings: Specify any combination of eight 0's and 1's that is used as the first switch setting for the submitted job.

DSPSBMJOB
Specifies whether the job being submitted is allowed to be shown on the Submitted Jobs Display. Any submitted job of the type specified by the SBMFROM parameter of the Work with Submit Job (WRKSBMJOB) command can be shown if the job is not prevented by this parameter.

***YES:** This job can be shown by the WRKSBMJOB command.

***NO:** This job is not shown on any display produced by the WRKSBMJOB command.

MSGQ
Specifies the qualified name of the message queue to which messages are sent.

Note: If a job ends abnormally, the online help information of the completion message sent specifies the possible causes.

***USRPRF:** A completion message is sent to the message queue specified in the user profile of the user who submits the job.

***WRKSTN:** A completion message is sent to the work station message queue of the work station from which the job was submitted. If the job is submitted by a batch job, no completion message is sent.

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

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

> ***LIBL:** All libraries in the job's library list are searched until the first match is found.
>
> ***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.
>
> *library-name:* Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which messages are sent.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons for this job.

***CURRENT:** The sort table specified for the job that is currently running is used.

***SYSVAL:** The system value QSRTSEQ is used.

***USRPRF:** The sort table specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***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 job.

LANGID

Specifies the language identifier to be associated with this job. The language identifier is used when *LANGIDUNQ or *LANGIDSHR is specified on the sort sequence prompt (SRTSEQ parameter). If the job CCSID is 65535, this parameter is also used to determine the value of the job default CCSID (DFTCCSID).

***CURRENT:** The language identifier specified for the job that is currently running is used.

***SYSVAL:** The system value QLANGID is used.

***USRPRF:** The language ID specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

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

CNTRYID

Specifies the country or region identifier to be used by the job.

***CURRENT:** The country or region identifier specified for the job that is currently running is used.

***SYSVAL:** The system value QCNTRYID is used.

***USRPRF:** The country or region ID specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

country-or-region-ID: Specify the country or region identifier to be used by the job.

CCSID

Specifies the coded character set identifier (CCSID) used for the submitted job.

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

***CURRENT:** The CCSID specified for the job that is currently running is used.

***USRPRF:** The CCSID specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***SYSVAL:** The CCSID specified for the system value QCCSID at the time the job is started is used.

***HEX:** The CCSID 65535 is used.

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

SBMFOR

Specifies the job name to be used on the SBMFROM parameter of the WRKSBMJOB command.

***CURRENT:** The name of the currently active job is used.

job-number/user-name/job-name: Specify the job number, user name, and job name to be used.

Note: You must have *JOBCTL authority to use this parameter.

JOBMSGQMX

Specifies the maximum size of the job message queue.

***JOBD:** The value specified in the job description determines maximum size of the job message queue.

***SYSVAL:** The value in QJOBMSGQMX (system value) at the time the job is started is used as the maximum size of the job message queue.

maximum-size-of-job-message-queue: Specify a value in the range of 2 to 64 megabytes.

JOBMSGQFL

Specifies the action that should be taken when the job message queue is full.

***JOBD:** The value specified in the job description determines the action that should be taken.

***SYSVAL:** The value specified for the QJOBMSGQFL system value is used.

***NOWRAP:** The message queue does not wrap when it is full. This action ends the job.

***WRAP:** The message queue wraps to the start of the message queue when it is full and starts filling the message queue again.

***PRTWRAP:** The message queue wraps the job message queue when full it is and prints the messages that are being overlaid because of wrapping.

CPYENVVAR

Specifies whether the environment variables from the submitting job are copied to the new job.

***NO:** The environment variables are not copied.

***YES:** The environment variables are copied.

ALWMLTTHD

Specifies whether or not the job is allowed to run with multiple user threads. This attribute does not prevent the operating system from creating system threads in the job. This attribute is not allowed to be changed after the job is submitted.

***JOBD:** The value specified in the job description determines whether or not the job is allowed to run with multiple user threads.

***NO:** The job is not allowed to run with multiple user threads.

***YES:** The job is allowed to run with multiple user threads.

INLASPGRP

Specifies the initial setting for the auxiliary storage pool (ASP) group name for the initial thread of the submitted job. A thread can use the Set Auxiliary Storage Pool Group (SETASPGRP) command to change its library name space. When an ASP group is associated with a thread, all libraries in the independent ASPs in the ASP group are accessible and objects in those libraries can be referenced using regular library-qualified object name syntax. The libraries in the independent ASPs in the specified ASP group plus the libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32) form the library name space for the thread.

***CURRENT:** The ASP group name for the current thread is used for the submitted job.

***JOBD:** The initial ASP group name specified in the job description is used for the submitted job.

***NONE:** Specifies the initial thread of the submitted job will be started with no ASP group. The library name space will not include libraries from any ASP group. Only the libraries in the system ASP and any basic user ASPs will be in the library name space.

auxiliary-storage-pool-group-name: Specify the name of the ASP group to be set for the initial thread of the submitted job. The ASP group name is the name of the primary ASP device within the ASP group. All libraries from all ASPs in this ASP group will be included in the library name space.

SPLFACN

Specifies whether or not spooled files are accessed through job interfaces after the job ends. Keeping spooled files with jobs allows job commands such as Work with Submitted Jobs (WRKSBMJOB) to work with the spooled files even after the job has ended. Detaching spooled files from jobs reduces the use of system resources by allowing job structures to be recycled when the jobs end.

***CURRENT:** The value from the current job is used for the submitted job.

***JOBD:** The value in the job description is used.

***SYSVAL:** The value specified in the system value QSPLFACN is used.

***KEEP:** When the job ends, the spooled files are kept with the job and the status of the job is updated to indicate that the job has completed.

***DETACH:** When the job ends, the spooled files are detached from the job and the job is removed from the system.

Examples for SBMJOB

Example 1: Submitting a Job

```
SBMJOB  JOB(SPECIAL)  JOBD(MYLIB/MYJOBD)
  CMD(CALL MYPROG)
```

This command causes the job named SPECIAL to be submitted. Most of the attributes for the job are taken from the job description MYJOBD, or the job that is currently running, except for the command. The CALL command is placed on the submitted job's message queue so that the program MYPROG can be called and run later.

Example 2: Submitting a Job

```
SBMJOB  JOB(PAYROLL)  JOBD(PAYROLL)  INQMSGRPY(*RQD)
```

This command submits a job named PAYROLL to the system. All the information needed for this job (such as the job queue and routing data but not the inquiry message control value) is contained in the job description PAYROLL, or the job that is currently running. The library list in effect for the job issuing this command is used to find the job description. All inquiry messages sent during running of this job requires the receiver of the inquiry message to reply.

Example 3: Submitting a Job to a Job Queue

```
SBMJOB  JOBD(*USRPRF)  JOB(COPY12)  JOBQ(NIGHTQ)
  CMD(CPYF FILEA FILEB)
```

This command submits the job COPY12, which uses the job description in the user profile of the submitting job, to the job queue NIGHTQ. The CMD parameter provides the CL command necessary for the job to run. A command such as this might be used to copy the file at night while the system is unattended.

Error messages for SBMJOB

***ESCAPE Messages**

CPF133A
SBMJOB not allowed during IPL.

CPF1338
Errors occurred on SBMJOB command.

CPF1651
Sort sequence table not accessed.

SBMJOBJS (Submit Job using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

SBMJOBJS Command syntax diagram

Purpose

The Submit Job using Job Scheduler (SBMJOBJS) command allows you to submit a job from Job Scheduler.

Note: When referring to a job in this command, we are referring to an entry in Job Scheduler. An **entry** in Job Scheduler is a user-defined name for commands or programs that you want to process at scheduled times and dates. Job Scheduler jobs (entries) are not OS/400 objects.

Restrictions:

1. The user must have use authority to the job description and the user profile.
2. The user must have use and add authorities to the message queue and the output queue.
3. The user must have read authority to the job queue and to all libraries associated with the specified objects. CHGJOBJS, SBMJOBJS
4. The user must have use authority to the *ADDJOB function.

Required Parameter

JOB Specifies the name of the job schedule entry that you want to submit. You must specify a job and alternately can specify a group to which the job belongs as well as a sequence number for the job within the group.

Submitting a job from within a group does not start the group. It only runs the job that was selected for submission. If the sequence 1 job within the group is submitted, the whole group is submitted.

Element 1: Job

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

Element 2: Group

***NONE:** There is not a group associated with this job.

group-name: Specify the name of the group of which this job is a member.

Element 3: Group sequence

***NONE:** There is not a sequence number assigned to the job.

group-sequence-number: Specify the sequence number of the job. Sequence numbers can range from 1 to 99.

Optional Parameters

TIME Specifies the time that you want this job to be submitted. Jobs can be submitted at a specified time, immediately, or at the next scheduled time. Times are entered in hour, minute (HHMM) format and can range from 0001 to 0024 (midnight).

***SCHED:** The job is to be submitted by Job Scheduler at the next scheduled time.

***IMMED:** The job is submitted immediately.

submit-time: Specify the time that you want the job to be submitted by Job Scheduler in hour, minute (HHMM) format.

DATE Specifies the date that you want this job to be submitted. Dates are specified in job date format.

***CURRENT:** The job is to be submitted by Job Scheduler on the current date at the time specified in the TIME parameter.

submit-date: Specify the date that you want the job to be submitted by Job Scheduler.

STRSEQ
Specifies the starting command sequence number for the job that you are submitting.

***FIRST:** Start with the first command sequence number for the job that you are submitting.

sequence-number: Specify the sequence number of the command within the job that you want to start with.

PARM Specifies the name of the parameter that you want to use for the submitted job. Parameters that you specify must be defined in Job Scheduler.

You can enter multiple values for this parameter. If you are on an entry display and you need additional entry fields to enter these multiple values, type a plus sign (+) in the entry field opposite the phrase “+ for more”, and press the Enter key.

Any character is valid for entry into this field. Validity will be checked to the extent of the use of the parameter within your programs or at the processing of the Job Scheduler job when submitted. The number of characters in the PARM parameter should correspond with the length placed in the **Parameter length** field when the parameter was added to Job Scheduler. Data entered in a shorter length than the **Parameter length** field will pass data padded with blanks to the length of the parameter. Leading blanks and embedded blanks will be passed to the parameter in your request data exactly as keyed. Apostrophes (’) used in parameter data must appear in pairs to be valid. If only one apostrophe is used, the parameter will be rejected with the error “Quotes (’’) in Parameter Data Must Appear In Pairs”. You must correct this error before continuing.

***NONE:** The job that you are submitting does not require any parameters entered here.

Element 1: Parameter name

parameter-name: Specify the parameter that you want to use with this job.

Element 2: Parameter data

Specify the parameter data that you want to use for the parameter name that you specified in the PARM parameter.

parameter-data: Specify the parameter data for the parameter name.

RMTLOCNAME

Specifies the location and network identification of the remote location name on which to run the job.

Note: A value specified in the RMTLOCNAME parameter will be ignored when used with schedule code *ALTERNATE.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** Use the remote location name specified in the job controls.

***LCL:** Run the job on the local server.

remote-location-name: Specify the name of the location associated with the server on which to run the job.

network-ID.location-name: Specify the network identifier and the name of the location associated with the server. Specify these values using the format nnnnnnnn.cccccccc where nnnnnnnn is the network identifier and cccccccc is the location name.

MAXRUN

Specifies the maximum run duration in minutes for the job.

Note: A value specified in the MAXRUN prompt will be ignored when used with schedule code *ALTERNATE.

***SAME:** The value specified in the Job Scheduler job entry is used.

***NOMAX:** There is no maximum duration for the job.

maximum-run-time: Specify the number of minutes that is the maximum duration for this job. After this number of minutes has passed, Job Scheduler will end the job whether it has completed or not. The maximum minutes can range from 1 to 9999 minutes.

PGRRCPNORM

Specifies the pager recipient who is to receive normal completion messages for the job that you are submitting. This field is used in conjunction with command specified in the CHGPGRJS command.

Note: A paging product must be installed before this feature may be used.

You can specify the pager message that you want to send to the specified recipient when the job completes normally. The values that you specify for Pager recipient normal and Pager message are the substitution values used for the &RCP and &MSGTXT variables respectively in the CHGPGRJS.

Element 1: Pager recipient normal

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** You are using the pager recipient specified in the Job Scheduler job controls.

***NONE:** No pager recipient is assigned to receive messages when this job completes normally.

recipient-name: Specify the name of a recipient who is to receive messages from the job when it completes normally.

Element 2: Pager message

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The pager recipient is sent the pager message from the Job Scheduler job controls.

***COMP:** The job completion of the job is sent.

pager-message: Specify the page message that you want to send to the pager recipient when this job completes normally.

PGRRCPABN

Specifies the pager recipient who is to receive abnormal completion messages for the job that you are submitting. This field is used in conjunction with the CHGPGRJS command.

Note: A paging product must be installed before this feature may be used.

You can specify the page message that you want to send to the specified recipient when the job completes abnormally. The values that you specify for Pager recipient abnormal and Pager message are the substitution values used for the &RCP and &MSGTXT variables respectively in the CHGPGRJS command.

Element 1: Pager recipient abnormal

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** You are using the pager recipient specified in the Job Scheduler job controls.

***NONE:** No pager recipient is assigned to receive messages when this job completes abnormally.

recipient-name: Specify the name of a recipient who is to receive messages from the job when it completes abnormally.

Element 2: Pager message

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The pager recipient is sent the page message from the Job Scheduler job controls.

***COMP:** The job completion of the job is sent.

pager-message: Specify the page message that you want to send to the pager recipient when this job completes abnormally.

ALTJOB

Specifies the name of the alternate job for the job. Alternate jobs only run when a regular job ends abnormally. Jobs are not required to have alternate jobs. You can also specify the group and sequence number for the alternate job that you specify.

***SAME:** The value specified in the Job Scheduler job entry is used.

***NONE:** The job does not have an alternate job.

Element 1: Alternate job

alternate-job-name: Specify the name of the alternate job that you want to use with the job that you are adding.

Element 2: Group

***NONE:** The alternate job does not have an alternate group.

group-name: Specify the name of the group associated with the alternate job for this job.

Element 3: Group sequence

***NONE:** The alternate job does not have a sequence number assigned.

group-sequence-number: Specify the sequence number assigned to the alternate job. Sequence numbers can range from 1 to 99.

RPTDSTID

Specifies the report distribution ID that is used to distribute the reports generated as a result of processing the job.

***SAME:** The value specified in the Job Scheduler job entry is used.

***NONE:** The job does not have a report distribution ID.

report-distribution-ID: Specify the report distribution ID that you want to associate with this job.

RCYACN

Specifies the recovery action to be taken if the job cannot be submitted at the designated time because the system is powered down or in restricted state. The action specified on the parameter occurs at the next IPL or when the Job Scheduler system becomes active.

Jobs submitted during IPL or when the system comes out of restricted state are submitted in the same order that they would have been had the jobs been submitted at the times specified in the job schedule entries. If multiple occurrences of a recurring job are missed, the job is submitted only once. The first missed occurrence of the job is calculated from the current date.

Since the scheduler portion of IPL need not be complete for the IPL of the system to be complete, other jobs may start on the system before all of the jobs have been submitted.

This parameter does not apply:

- When a job is released after being held at the date and time it was to be submitted.

Note: A value specified in the RCYACN prompt will be ignored when used with schedule code *ALTERNATE, *DEPJOB and *GROUP (subordinate job group).

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The job uses the recovery action specified in the Job Scheduler job controls.

***SBMRLS:** The job is submitted in release state (RLS).

***SBMHLD:** The job is submitted in the held state (HLD).

***NOSBM:** The job is not submitted.

Specifying *NOSBM affects only missed occurrences of the job. If the job schedule entry is a recurring job, future occurrences are not affected.

JOBD Specifies the name of the job description used with this job.

Element 1: Job Description

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The job description from the Job Scheduler job controls is used for this job.

***USRPRF:** The job description in the user profile under which the submitted job runs is used as the job description of the submitted job.

job-description-name: Specify the name (library-name/job-description-name) of the job description used for the job.

Element 2: Library

***LIBL:** The library list is used to locate the job description name.

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

library-name: Specify the name of the library where the job description name is located.

JOBQ Specifies the name of the job queue in which this job is placed.

Element 1: Job Queue

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The job queue specified in the Job Scheduler job controls is used for this job.

***JOBD:** The submitted job is placed on the job queue named in the specified job description.

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

Element 2: Library

Specify the name (library-name/job-queue-name) of the job queue on which the submitted job is placed.

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

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

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

JOBPTY

Specifies the job queue scheduling priority. Valid values range from 1 through 9, where 1 is the highest priority and 9 is the lowest priority.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The scheduling priority default specified in the Job Scheduler job controls is used for the job.

***JOBD:** The scheduling priority specified in the job description is used for the job.

job-priority: Specify a value, ranging from 1 through 9, for the scheduling priority for the job.

OUTPTY

Specifies the output queue priority for spooled output files that are produced by this job. Valid values range from 1 through 9, where 1 is the highest priority and 9 is the lowest priority.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The output priority default specified in the Job Scheduler job controls is used for the job.

***JOBD:** The output priority specified in the job description is used for the job.

output-priority: Specify a value, ranging from 1 through 9, for the output priority for the job.

PRTDEV

Specifies the qualified name of the default printer device for this job.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The printer specified in the Job Scheduler job controls is used by the job as the printer device.

***USRPRF:** The printer device specified in the user profile where the submitted job runs is used as the printer device for this job. The printer device name is obtained from the profile when this job is submitted.

***SYSVAL:** The printer device specified in the system value, QPRTDEV, when this job is submitted is used.

***JOBD:** The printer device specified in the job description is used for the submitted job.

printer-device-name: Specify the name of the printer device used for the submitted job.

OUTQ Specifies the qualified name of the output queue that is used for spooled output produced by the job. This parameter only applies to spooled printer files that specify *JOB for the output queue.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The output queue specified in the Job Scheduler job controls is used as the job's output queue.

***JOBD:** The output queue named in the job description used with the submitted job is the job's default output queue.

output-queue-name: Specify the name (library-name/output-queue-name) of the output queue that is used as the default output queue by the submitted job.

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

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

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

USER Specifies the name of the user profile for the job being submitted. If *RQD is specified in the job description, *JOBD cannot be specified; a user name must be specified instead.

Note: The following IBM-supplied objects are not valid on this parameter:

- QDBSHR
- QDFTOWN
- QDOC
- QLPAUTO
- QLPINSTALL
- QRJE
- QSECOFR
- QSPL
- QSYS
- QTSTRQS

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The user profile specified in the Job Scheduler defaults is used for the job being submitted.

***CURRENT:** The same user profile used by the job that is currently running is used for the submitted job.

***JOBD:** The user profile named in the specified job description is used for the job being submitted.

user-name: Specify the name of the user profile that is used for the job being submitted. You must be authorized to the user profile; the user profile must be authorized to the job description.

PRTTXT

Specifies up to 30 characters of text that is printed at the bottom of each page of printed output and on separator pages.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job controls is used for this job.

***JOBD:** The value in the job description is used for this job.

***SYSVAL:** The system value, QPRTTXT, is used for this job.

***BLANK:** No text is printed.

print-text: Specify the character string that is printed at the bottom of each page. A maximum of 30 characters can be entered, enclosed in apostrophes. The text on the listing will be centered in the same way it is entered.

RTGDTA

Specifies the routing data used to start the first routing step in the job. The routing data is used to determine the routing entry that identifies the program that the job runs.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job controls for routing data is used for this job.

***JOBD:** The routing data specified in the job description is used to start the routing steps.

routing-data: Specify the character string that is used as routing data for the job. A maximum of 80 characters can be entered, enclosed in apostrophes if necessary.

CURLIB

Specifies the name of the current library associated with the job being run.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The Job Scheduler job controls are used for the submitted job.

***USRPRF:** The current library in the user profile where the submitted job runs is used as the current library for the submitted job.

***CRTDFT:** There is no current library for the submitted job. If objects are created in the current library, QGPL is used as the default current library.

current-library-name: Specify the name of a library used as the current library of the submitted job.

LIBL Specifies the name of the library list that is used to search for any operating system object names that were specified without a library qualifier. If you want to select a library list from a list, place the cursor within the LIBL parameter field and press F4.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBD:** The library list in the job description used with this job is used as the initial user part of the library list for the job.

***SYSVAL:** The system default user library list is used by this job. It contains the library names that were specified in the system value, QUSRLIBL, at the time that the job is started.

***NONE:** The user portion of the initial library list for this job will be empty.

***JOBCTL:** The Job Scheduler job control is used for the library list.

library-list-name: Specify the name of the library list that you want to use for this job.

LOG Specifies the message logging values used to determine the amount and type of information sent to the job log by this job. This parameter has three elements: the message (or logging) level, the message severity, and the level of message text. If no values are specified on this parameter, the values specified in the job description associated with this job are used.

Element 1: Message level

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value specified in the Job Scheduler job controls for logging is used for this job.

***JOBD:** The value specified for message logging in the job description is used for this job.

message-level: Specify a value, ranging from 0 to 4, that specifies the message logging level used for this job's messages. The possible logging levels are:

0: No data is logged.

1: The following information is logged: All messages sent to the job's external message queue with a severity level greater than or equal to the message severity specified (this includes the indications of job start, job end and job completion status).

2: The following information is logged:

- Logging level 1 information
- Requests or commands being logged from a CL program for which messages are issued with a severity code greater than or equal to the severity level specified.

- All messages associated with a request, or commands being logged from a CL program, that results in a high-level message with a severity level greater than or equal to the severity specified.

3: The following information is logged:

- Logging level 1 information
- All requests or commands being logged from a CL program.
- All messages associated with a request, or commands being logged from a CL program, that results in a high-level message with a severity level greater than or equal to the severity specified.

4: The following information is logged:

- All requests or commands being logged from a CL program and all messages with a severity code greater than or equal to the severity specified, including trace messages.

  Note: A high-level message is one that is sent to the program message queue of the program that received the request or commands being logged from a CL program.
- Message severity

  ***SAME:** The value specified in the Job Scheduler job entry is used.

  ***JOBCTL:** The value specified in the Job Scheduler job controls for message severity is used for this job.

  ***JOBD:** The value specified for message logging in the job description is used for this job.

  message-severity: Specify a value, ranging from 00 to 99, that specifies the lowest severity level that causes an error message to be logged in the job's log.
- Message text

  ***SAME:** The value specified in the Job Scheduler job entry is used.

  ***JOBCTL:** The value specified in the Job Scheduler job controls for message text is used for this job.

  ***JOBD:** The value specified for message logging in the job description is used for this job.

  ***MSG:** Only message text is written to the job's log or shown to the user.

  ***SECLVL:** Both the message text and message help of the error message is written to the job's log or shown to the user.

  ***NOLIST:** No job log is produced if the job competes normally. If the job ends abnormally (if the end of job code is 20 or higher), a job log is produced. The messages appearing in the job's log contain both message text and online help information.

LOGCLPGM

Specifies whether the commands that are run in a control language program are logged to the job log by way of the CL program's message queue. This parameter sets the status of the job's logging flag. If *JOB has been specified for the Message logging prompt (LOG parameter) in the Create CL Program (CRTCLPGM) command, the flag set in the LOGCLPGM parameter Log CL program commands prompt (LOGCLPGM parameter) is used. Other values for the Message logging prompt (LOG parameter) override the Log CL program commands prompt (LOGCLPGM parameter). The commands are logged in the same manner as the requests.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job controls is used for this job.

***JOBD:** The value specified in the job description is used.

***NO:** The commands in a CL program are not logged to the job log.

***YES:** The commands in a CL program are logged to the job log.

INQMSGRPY

Specifies the way that predefined messages that are sent as a result of running this job are answered. You can specify that no change is made in the way that predefined messages are answered, or that all inquiry messages require a reply, or that a default reply is issued, or that the system reply list is checked for a matching reply as each predefined inquiry message is sent.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job control for inquiry message reply is used for this job.

***JOBD:** The inquiry message reply control specified in the job description used with this job is used.

***RQD:** A reply is required by the receiver of the inquiry message for all inquiry messages that occur when this command is run.

***DFT:** The default message reply is used to answer any inquiry messages that occur when this command is run.

***SYSRPYL:** The system reply list is checked to see if there is an entry for any inquiry message that is issued as a result of running this job that has a message identifier and any comparison data that match the inquiry message identifier and message data. If a match occurs, the reply value in that entry is used. If no entry exists for that message, a reply is required.

HOLD Specifies whether this job is held at the time that it is put on the job queue. A job placed on the job queue in the hold state is held until it is released by the Release Job (RLSJOB) command or ended, either by the End Job (ENDJOB) command or by the Clear Job Queue (CLRJOBQ) command.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job control for hold on job queue is used for this job.

***JOBD:** The value specified in the job description determines whether the job is held when it is put on the job queue.

***NO:** The job is not held when it is put on the job queue.

***YES:** The job is held when it is put on the job queue until it is released by a Release Job (RLSJOB) command or ended by an End Job (ENDJOB) command.

SWS Specifies the first settings for a group of eight job switches used with this job. These switches can be set or tested in a CL program and used to control the flow of the program. Only 0’s (off) and 1’s (on) can be specified in the 8-digit character string.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job control for job switches is used for this job.

***JOBD:** The value specified in the job description is the first setting for the job’s switches.

switch-settings: Specify any combination of eight zeros and ones that is used as the first switch setting for the submitted job.

MSGQ

Specifies the name of the message queue to which a completion message is sent when the submitted job has completed running, either normally or abnormally. If an abnormal ending occurs, the help information for the completion message specifies the possible causes.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The value in the Job Scheduler job control for message queue is used for this job.

***USRPRF:** A completion message is sent to the message queue specified in the user profile of the submitter.

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

message-queue-name: Specify the name (library-name/message-queue-name) of the message queue on which the completion message is to be sent.

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

***CURLIB:** The current library is used to locate the message queue name.

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

ACGCDE
Specifies the accounting code that is used when logging system resource use for jobs that use this description.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The accounting code for jobs using this description is obtained from the job controls.

***JOBD:** The accounting code for jobs using this description is obtained from the job description.

***USRPRF:** The accounting code for jobs using this description is obtained from the user profile associated with the job.

***BLANK:** An accounting code of 15 blanks is assigned to jobs that use this description.

accounting-code: Specify the accounting code that you want to use for jobs using this description.

RUNPTY
Specifies the run priority for the job. Run priority is a value ranging from 1 (highest priority) through 99 (lowest priority), that represents the importance of the job when it competes with other jobs for machine resources. This value represents the relative (not absolute) importance of the job. If the job is rerouted, this value is reset according to the job’s importance within the job class.

***SAME:** The value specified in the Job Scheduler job entry is used.

***JOBCTL:** The run priority is obtained from the job controls.

***NOCHG:** The run priority is not changed when job processing starts.

machine-running-priority: Specify the run priority, ranging from 1 through 99, that the job uses.

Example for SBMJOBJS

Example 1: Submitting a Job

```
SBMJOBJS JOB(JOB02) TIME('300') DATE('8/19/99')
```

In this example JOB02 is scheduled for submission at 3:00 a.m. on August 19, 1999. DATE is expressed in mm/dd/yy format in this example.

Error messages for SBMJOBJS

None

SBMNETJOB (Submit Network Job) Command Description

SBMNETJOB Command syntax diagram

Purpose

The Submit Network Job (SBMNETJOB) command sends an input stream to another system user on the SNADS network. (The input stream is sent to another user where it can be filed, submitted, or rejected.) When the input stream arrives, its placement is governed by the job action (JOBACN) network attribute. If the value of JOBACN is *SEARCH, the entry in the network job table at the receiving system is used to determine the action taken. (The network job table entry is determined by the value specified on the ACTION parameter on the Add Network Job Entry (ADDNETJOBE) or Change Network Job Entry (CHGNETJOBE) commands.) At the receiving system, the job may be submitted immediately, filed for placement by the receiving user, or rejected.

When the input stream arrives at the destination system, a message is sent to both the recipient of the input stream as well as the originator of the input stream stating that the input stream arrived.

This command can only be used to send an input stream to a user on a remote system.

Restrictions:

1. To use this command, the user must have object operational and read authority to the file that is submitted, and for the library that contains the file.
2. The user must be enrolled in the system distribution directory to use this command. (For information on enrolling in the system distribution directory, see the SNA Distribution Services 📖 book.)
3. If the job action (JOBACN) network attribute on the receiving system is set to *SEARCH, there must be an entry for the user in the network job table on the receiving system. The entry in this table specifies a user profile on the receiving system that is used to verify that the user is authorized to submit the job on that system. The user profile on the receiving system must be authorized to use the job queues, and must have object operational authority for the job descriptions specified by the JOB commands in the input stream.
4. The file that is submitted cannot contain more than approximately 2 billion bytes of data.

Required Parameters

FILE Specifies the name of the physical file containing the input stream that is sent.

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

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

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

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

file-name: Specify the name of the physical file that contains the input stream being sent.

TOUSRID

Specifies the two-part user ID of one or more users to whom the input stream is submitted, or the name of one or more distribution lists containing the two-part user IDs of one or more users to whom the file is to be sent. A combination of both user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required.

Notes:

1. Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Directory Entries (DSPDIRE) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.
2. The user specified in this parameter, or in the distribution list, must be a remote user. The SBMNETJOB command cannot be used to send input streams to local users.

Optional Parameters

MBR Specifies the member that is sent from the file.

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

member-name: Specify the name of the file member that is submitted.

PTY Specifies the queuing priority used for the input stream when it is being routed through a SNADS network.

***NORMAL:** The input stream is sent with a service level priority of data low, which is used for most data traffic. On an iSeries 400, data low distributions are placed on the normal distribution queue specified for the route.

***HIGH:** The input stream is sent with a service level priority of data high, which is used for high priority data traffic. On an iSeries 400, data high distributions are placed on the data high distribution queue specified for the route.

Example for SBMNETJOB

```
SBMNETJOB   FILE(PAYROLL)  TOUSRID(PAYROLL SYSTEM1)
  MBR(WEEKLY)
```

This command sends the input streams contained in member WEEKLY of file PAYROLL to user ID PAYROLL SYSTEM1.

Error messages for SBMNETJOB

*ESCAPE Messages

CPF8056
File &1 in &2 not a physical file.

CPF8058
File &1 is a spooled file.

CPF8063
Cannot assign necessary resource.

CPF8065
Input stream &1 in &2 member &3 not sent to any users.

CPF8066
One or more user identifiers on this command is not correct.

CPF8068
Error detected while processing file to be sent.

CPF8072
Object to be sent is greater than maximum size of 2GB.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9830
Cannot assign library &1.

CPF9845
Error occurred while opening file &1.

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

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

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

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

SBMNWSCMD (Submit Network Server Command) Command Description

SBMNWSCMD Command syntax diagram

Purpose

The Submit Network Server Command (SBMNWSCMD) command submits a command to run on the designated server. For SVRTYPE(*WINDOWSNT), command output is directed as specified by the CMDSTDOUT parameter. For all other types, command output is directed to the job log of the job that issues this command.

Restriction: Users must have *JOBCTL special authority to use this command.

Required Parameters

CMD Specifies the command submitted to the network server. The syntax of the command string specified for this parameter is not checked.

***NOLOGCMD:** Specifies that the user wants to submit a command string that is not logged in the joblog. This is useful if the command string contains sensitive data such as passwords. If *NOLOGCMD is specified, a command string must be entered on the NOLOGCMD parameter.

command: The command string to be submitted to the network server. This command string will be shown in the joblog.

SERVER
Specifies the name of the server to which the command is submitted.

Optional Parameters

NOLOGCMD
Specifies the command string to be submitted to the network server. This command string will not be echoed to the joblog. Use this parameter to submit commands that contain sensitive data such as passwords. This parameter is required if, and only allowed when, CMD(*NOLOGCMD) is specified.

Note: Since the command is being executed on a server, the presence of this option does not prevent the server from returning the command string as part of the output data. Thus, depending on the command, it may still be returned and displayed in the job log or spool file.

SVRTYPE
Specifies the type of server to which the command is sent.

***NWSUSRA:** The server type specified in the network server user attributes (CHGNWSUSRA command) for the user profile running the SBMNWSCMD is used.

***NWSA:** The server type specified in the system network server attributes (CHGNWSA command) is used.

***NETWARE:** The server type is NetWare. Only NetWare commands will run on a NetWare server.

***WINDOWSNT:** The server type is Windows NT. Only Windows NT commands will run on a Windows NT server.

CMDTYPE
Specifies the type of command string specified in the CMD parameter. This is used by the system to determine what type of processing needs to occur for the command string.

***SVRTYPE:** The command string is processed based on the type of server specified in the SVRTYPE parameter. If the server is a NetWare server, the command string is assumed to be a NetWare command. If the server is a *WINDOWSNT server, the command string is assumed to be a Windows NT command.

***NETWARE:** The command string is a NetWare command.

***WINDOWSNT:** The command string is a Windows NT command.

CMDSTDOUT
Specifies where the standard output returned from the command is to be stored if any exists. Standard output can be written to the job log of the job that issues this command, it can be written to a spooled file, or it can be written to a file. The standard error returned from the command will always be directed to the job log of the job that issues this command if any exists.

Note: This parameter is only valid when SVRTYPE(*WINDOWSNT) is specified.

***JOBLOG:** The standard output of the network server command will be directed to the job log of the job that issues this command. It shares the job log with the standard error output of the network server command. Both may be mixed in the job log, depending on the order by which the command writes standard output and standard error information.

***PRINT:** The standard output of the network server command will be directed to a spooled file. Certain control characters such as line feeds and carriage returns are converted to new lines and other non-displayable control characters such as highlight and underscore are converted to blanks.

'file-path-name': Specify the path name of the file to which the standard output of the network server command will be directed. The specified path must exist. If the file doesn't exist, it will be created. If the file exists, all data will be overlaid. Additional information about path name is in the Integrated File System Introduction topic in the Information Center.

CVTSTDOUT
Specifies whether the standard output will be converted from the server's code set to the CCSID of the OS/400 user profile that submitted the command. For binary output, CVTSTDOUT(*NO) should be specified.

Note: This parameter is only valid when CMDSTDOUT(*PRINT) or CMDSTDOUT(file-path-name) is specified.

***YES:** The output will be converted from the server's code set to the CCSID of the OS/400 user profile that submitted the command.

***NO:** The output will not be converted from the server's code set.

AUTDMN
Specifies the Windows NT domain where the user is authenticated.

Note: This parameter is only valid when SVRTYPE(*WINDOWSNT) is specified.

***PRIMARY:** The user is authenticated on the primary domain of the server.

***LOCAL:** The user is authenticated on the local server.

'domain-name': Specify the domain name where the user is authenticated.

Examples for SBMNWSCMD

Example 1: Submitting a NetWare command

```
SBMNWSCMD   CMD('CONFIG')  SERVER(NTW01)  SVRTYPE(*NETWARE)
```

This command submits the NetWare CONFIG command to run on the server named NTW01. Output is returned to the job log.

Example 2: Submitting a Windows NT 'net config server' command

```
SBMNWSCMD   CMD('net config server')
            SERVER(NTSVR)  CMDSTDOUT(*JOBLOG)
```

This command will display the Windows NT Server service settings on the Windows NT server NTSRVR. Standard output from the command is returned to OS/400 and directed to the job log.

Error messages for SBMNWSCMD

***ESCAPE Messages**

CPFA43F
Network server command not submitted.

CPFA46C
Unable to complete command processing on server &1.

CPFA46F
Network server description &1 not found.

SBMRMTCMD (Submit Remote Command) Command Description

SBMRMTCMD Command syntax diagram

Purpose

The Submit Remote Command (SBMRMTCMD) command submits a command via Distributed Data Management (DDM) for running on the target system specified by a DDM file. The value for the RMTLOCNAM parameter in the DDM file determines the communications line used, and thus indirectly identifies the target system that receives the submitted command.

This command is used to send only CL commands to another iSeries 400. It is valid only when the target system is an iSeries 400 or a System/38. It can be used for iSeries 400 or System/38 commands only; for example, Operation Control Language (OCL) commands cannot be sent to a target System/36.

The primary purpose of this command is to allow a source system user or program to perform file management operations and file authorization activities on files located on a target iSeries 400. More information on file management operations is in the Distributed Data Management topic in the Information Center. If the source system program or user has the correct authority, the following actions are examples of what is performed on remote files by using the SBMRMTCMD command:

- Create or delete physical, logical, device, or source files
- Grant or revoke object authority to remote files
- Check, rename, or move files or other objects
- Save or restore files or other objects

Although the command is used to do many things with files or objects, some are not as useful as others. For example, it can be used to show the file descriptions or field attributes of remote files or to dump files or other objects, but the output results remain at the target system. To display remote file descriptions and field attributes at the source system, use the DSPFD and DSPFFD commands, specifying SYSTEM(*RMT).

A secondary purpose of this command allows a user to perform nonfile operations, such as creating a message queue, or to submit user-written commands for running on the target system.

Restrictions:

1. Although remote file processing is synchronous in the user process, which includes two separate jobs (one running on each system), file processing on the target system operates independent of the source system. Commands such as OVRDBF, OVRMSGF, and DLTOVR, which are dependent on a specific recursion level or request level, may *not* function as expected.
2. Output (such as spooled files) generated by a submitted command exists only on the target system. The output is *not* sent back to the source system.
3. Some types of CL commands should *not* be submitted to a target iSeries 400. The following are examples of types that are *not* the intended purpose of the SBMRMTCMD command and that may produce undesirable results.

- All OVRxxxF commands that refer to device files, communications files, message files, or save files.
- The DSPxxxx commands should not be submitted because the output results remain at the target system.
- Commands that are valid only in an interactive environment, like WRKxxx or STRxxx.
- Job-related commands like RRTJOB that are used to control a target system's job. The CHGJOB command, however, *can* be used.
- Commands that are used to service programs, like SRVJOB, TRCJOB, TRCINT, or DMPJOB.
- Commands that may cause inquiry messages to be sent to the system operator, like STRPRTWTR or CPYTODKT. Pass-through can be used instead.

4. Translation is not performed for any *impromptu* messages caused by target system errors, because they are not stored on the system; the text for an impromptu message is sent directly to the source system shown. The message identifier of all other message types generated on the remote system is sent back to the source system. If the message text that exists for the message identifier on the source system has been translated, it will then be shown.
5. Up to 10 messages, generated during the running of a submitted command, can be sent by the target system to the source system. If more than 10 messages are generated, an additional *informational* message is sent that indicates that the messages exist in the job log for the target job on the target system. If one of those messages is an *escape* message, the first nine messages of other types are sent, followed by the informational message and the escape message.
6. In multithreaded jobs, this command is not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.

Required Parameters

CMD Specifies a character string of up to 2000 characters that represents a command that is run on the target system. The command must be allowed in both *BATCH and *EXEC (QCAEXEC) environments on the target iSeries 400.

The command must be enclosed in apostrophes if it contains embedded blanks or special characters.

Note: The normal rule of pairing apostrophes in quoted strings on the local system must be *doubled* when the same string is submitted to a remote system on this CMD parameter; this is required because the user is coding a quoted string within another quoted string. Therefore, when this parameter is being coded, wherever a single apostrophe would normally be paired with another apostrophe, *each occurrence in the outside set of apostrophes must be doubled* to produce the same results at the target system.

DDMFILE

Specifies the qualified name of the DDM file that is used to submit the command to the target system. If no library qualifier is given, *LIBL is used to find the file. The DDM file is used only to determine the remote location representing the target system. The remote file name associated with the DDM file is ignored by the SBMRMTCMD command.

The name of the DDM 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.

DDM-file-name: Specify the name of the DDM file used to submit the command to the target system.

Examples for SBMRMTCMD

Example 1: Deleting a File

```
SBMRMTCMD  CMD('DLTF LIBX/FRED')  DDMFILE(DENVER)
```

This command deletes the file named FRED in library LIBX on the target system that is associated with the DDM file named DENVER.

Example 2: Creating a Physical File

```
SBMRMTCMD  CMD('CRTPF  SALES/CAR  SRCFILE(QGPL/QDDSSRC)
  SRCMBR(MASTER)')  DDMFILE(DENVER)
```

This command creates the physical file CAR in library SALES using the data description specifications (DDS) in the source file QDDSSRC and source member named MASTER in the QGPL library. The DDS must already exist on the target system identified by the DDM file named DENVER in the target job's library list.

Example 3: Changing the Text Description

```
SBMRMTCMD  CMD('CHGDDMF  FILE(LIBX/STANLEY)
  TEXT(''Don''''t forget to pair apostrophes.'')')
  DDMFILE(SMITH)
```

This command changes the text in the description of the DDM file named STANLEY which is stored in library LIBX. Because the submitted command requires an outside set of single apostrophes (for the CMD parameter), each single or double apostrophe normally required in the TEXT parameter for *local* system processing must be doubled for *remote* system processing. The coding above produces a single apostrophe in the text when it is shown or printed on the remote system.

Example 4: Creating a DDM File

```
SBMRMTCMD CMD('CRTDDMF FILE(SALES/MONTHLY)
  RMTFILE(*NONSTD ''CAR.SALES(JULY)''
  RMTLOCNAME(DALLAS)') DDMFILE(CHICAGO)
```

This command creates (on the target system identified by the information in the DDM file named CHICAGO) another DDM file named MONTHLY. The new DDM file is stored in a library named SALES on the CHICAGO system. The new DDM file on the CHICAGO system is used to access a file and *member* on a different system named DALLAS. The accessed file is named SALES/CAR and the member name in the file is JULY.

Note that this CRTDDMF command string contains *three* sets of single apostrophes: one set to enclose the entire command being submitted, and a double set to enclose the file and member named in the RMTFILE parameter. This is how any IBM iSeries 400 file *member* name must be specified on the SBMRMTCMD command, because of the parentheses needed to enclose the member name.

Example 5: Replacing a Portion of the Library List

```
SBMRMTCMD  CMD('RPLLIBL  LIBL(QGPL QTEMP SALES EVANS)')
  DDMFILE(EVANS)
```

This command replaces the user's portion of the library list being used by the target job associated with the DDM file named EVANS, which is being used by the source job in which this SBMRMTCMD command is being submitted. In that source job, if there are other open DDM files that specify the same device and mode, this library list is used for them also.

Additional Considerations

Override example

The DDMFILE parameter is used to determine the target system to which the command is sent. Overrides that apply to the DDM file (not the remote file) are taken into account for this function. For example, if a file override was in effect for a DDM file because of the following commands, which override FILEA with FILEX, then the target system to which the DLTF command is sent is the one associated with the remote location values specified in FILEX (the values point to the DENVER system, in this case).

```
CRTDDMF  FILE(SRCLIB/FILEA)  RMTFILE(SALES/CAR)
  RMTLOCNAME(CHICAGO)

CRTDDMF  FILE(SRCLIB/FILEX)  RMTFILE(SALES/CAR)
  RMTLOCNAME(DENVER)

OVRDBF   FILE(FILEA)  TOFILE(SRCLIB/FILEX)
  SBMRMTCMD  CMD('DLTF RMTLIB/FRED')
  DDMFILE(SRCLIB/FILEA)
```

Distributed Data Management conversations: When a SBMRMTCMD command runs on the target system, it has a target system job associated with it. Successive SBMRMTCMD commands that are submitted using the same DDM file and DDM conversation may run in the same or different target system jobs, depending on the DDMCNV job attribute value. The DDMCNV job attribute value determines whether the DDM conversation is dropped (DDMCNV(*DROP)) or remains active (DDMCNV(*KEEP)) when the submitted function has completed. If the conversation is dropped, the next SBMRMTCMD command runs using a different target job.

Command syntax checking: The syntax of the command character string being submitted by the CMD parameter is not checked by the source system. For example, in the case of a user-defined command, the command definition object may not exist on the source system.

Command running results: Because the submitted command runs as part of the target system's job, the attributes of that job (such as the library search list, user profile, wait times, and running priority) may cause a different result than if the command were run locally.

Error message handling:

- For errors detected by the target system when processing the command that was submitted, the source system attempts to send to the source system user the same error information that the target system sent. However, if the source system does not have an equivalent message for the one sent by the target system, source system default message text (stating that the message does not exist on the source system) is sent to the source system user, along with the message identifier and message type of the actual target system message. The target system messages can be viewed on the source system by using pass-through and either the DSPJOB or DSPJOBLOG command.
- If the SBMRMTCMD command is used to call a CL program on the target system, unmonitored escape messages generated by the program are changed into inquiry messages and sent to the system operator. If the user does not want the target system operator to have to respond to this inquiry message before the job can continue, the user can refer to the following commands and do either of the following on the target system:
  - If the user wants to specify a default reply for a specific job, the INQMSGRPY parameter on either the CRTJOBD or CHGJOBD command can be used to specify either *DFT or *SYSRPYL in the job description for the target job.

– If the user wants to specify a default reply message for a specific inquiry message, the ADDRPYLE command can be used (on the target system) to add an entry to the system-wide automatic message reply list for that message.

Error messages for SBMRMTCMD

***ESCAPE Messages**

CPF9164
Target system does not support SBMRMTCMD.

CPF9165
File &1 in library &2 not a DDM File.

CPF917A
Error occurred on distributed file.

CPF917B
Target system &3 not available.

CPF9172
SBMRMTCMD command ended abnormally.

CPF9174
Error on call to user exit program on target system.

CPF9175
Error during processing of user exit program.

CPF9177
User exit program did not complete successfully.

CPF9178
Processing of the command specified by SBMRMTCMD failed.

CPF9182
Cannot start DDM communications.

TRCASPBAL (Trace ASP Balance) Command Description

TRCASPBAL Command syntax diagram

Purpose

The Trace ASP Balance (TRCASPBAL) command controls the function that gathers the auxiliary storage pool (ASP) usage statistics. The trace function monitors the frequency that data is accessed on the disk units within the specified ASP. The high-use data and the low-use data on the units is identified. The tracing of the usage of data on the units can be started on a specific ASP or for multiple ASPs. The trace may be started for a specific length of time. The trace can be stopped by specifying the *OFF value for the SET parameter (TRCASPBAL SET(*OFF)). The trace can be ended at any time and restarted at a later time. The statistics that are collected are cumulative. For example, if the trace is started and ended and then restarted without clearing the statistics, the second group of statistics are added to the first collection.

After statistics have been collected the ASP may be balanced using the Start ASP Balance (STRASPBAL) command, specifying TYPE(*USAGE) or TYPE(*HSM). After the balance has run to completion the statistics will be cleared automatically by the balance function.

The balancing of the ASP should be done shortly after the statistics have been collected. The usefulness of the balance is diminished as the trace statistics age. If the collection becomes aged, the statistics may be cleared using TRCASPBAL SET(*CLEAR).

A message will be sent to the system history (QHST) log when the trace function is turned on, when it is stopped, or the trace data is cleared.

For more information about ASP balancing see the Hierarchical Storage Management book.

Restrictions: You must have *ALLOBJ special authority to use this command. »

Required Parameter

SET Specifies what should be done.

***ON:** The tracing of the statistics will start.

***OFF:** The tracing of the statistics will be ended.

***CLEAR:** The statistics for the specified ASP will be cleared.

Optional Parameters

ASP Specifies the auxiliary storage pool (ASP) for which the ASP tracing function will be started, ended, or cleared. A value must be specified for the ASP parameter or the ASPDEV parameter.

***ALL:** ASP tracing will be started, ended, or cleared for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

auxiliary-storage-pool-number: Specify the ASP for which ASP tracing is to be started, ended, or cleared. Valid ASP numbers are 1 to 32. Up to 32 ASP numbers may be specified.

ASPDEV
Specifies the name of the auxiliary storage pool (ASP) device for which the ASP tracing function will be started, ended, or cleared. A value must be specified for the ASP parameter or the ASPDEV parameter.

***ALLAVL:** ASP tracing will be started, ended, or cleared for all ASP devices that currently have a status of 'Available'.

auxiliary-storage-device-name: Specify the name of the independent ASP device for which ASP tracing is to be started, ended, or cleared. Up to 32 ASP device names may be specified. «

TIMLMT
Specifies the amount of time, in minutes, that the ASP trace function will be allowed to run. When the time limit is reached the function will end. The trace function will not run across an IPL. » A time limit must be specified if *ON is specified for the SET parameter. «

time-limit: Specify the time limit that the trace function will be allowed to run. Valid values range from 1 to 9999 minutes.

Examples for TRCASPBAL

Example 1: Start Trace for ASP 1

```
TRCASPBAL  ASP(1) SET(*ON) TIMLMT(9999)
```

This command allows the user to start the ASP tracing function for ASP 1. This function will run until the user ends the trace or 9999 minutes have passed.

Example 2: End Tracing for All ASPs

```
TRCASPBAL  ASP(*ALL) SET(*OFF)
```

This command allows the user to end the ASP tracing function for each ASP that currently has a trace running.

Example 3: Clear the Trace Data for ASP 1

```
TRCASPBAL  ASP(1) SET(*CLEAR)
```

This command allows the user to clear the trace data for ASP 1. »

Example 4: End Tracing for All ASP Devices

```
TRCASPBAL  ASPDEV(*ALLAVL) SET(*OFF)
```

This command allows the user to end the ASP tracing function for each ASP device that currently has a trace running.

Error messages for TRCASPBAL

***ESCAPE Messages**

CPF1890
*ALLOBJ authority required for requested operation.

CPF18A9
ASP tracing for ASP &2 already started.

CPF18AA
ASP tracing not active for ASP &2.

CPF18AD
ASP &2 must contain more than a single unit.

CPF18AE
ASP &2 does not contain trace data.

CPF18B1
Trace function currently running for ASP &2.

CPF18B2
Balance function running for ASP &2.

CPF9829
Auxiliary storage pool &2 not found.

«

TRCCNN (Trace Connection) Command Description

TRCCNN Command syntax diagram

Purpose

The Trace Connection (TRCCNN) command allows the tracing of encrypted data flowing over internet protocol (IP) and Secure Sockets Layer (SSL) connections. Specific types of traces are started and stopped by using this command.

TRCCNN uses the Trace Internal (TRCINT) command to collect the trace records and generate an intermediate spooled file named QPCSMPRT. The QPCSMPRT spooled file data is used to generate a spooled file named QSYSPRT. The user data for the QSYSPRT file is 'TRCCNN'.

You can also use TRCCNN with a QPCSMPRT spooled file generated by using TRCINT directly. TRCCNN can extract and format the IP and SSL connection-related trace records. This allows you to use TRCINT to collect many types of trace records and then use TRCCNN to format the subset of trace records related to IP or SSL connections.

Restrictions:

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

Required Parameter

SET Specifies whether tracing is started, stopped or ended. Also, you can select to format trace record data collected previously using the TRCCNN or TRCINT (Trace Internal) command.

ON:** » The collection of internal trace records is started for the trace types specified in the TRCTYPE parameter. If *GEN is specified in the TRCTBL parameter then the trace table name will be **QTRCCNNxxxxxx* where *xxxxxx* is the job number of the current job. Otherwise the trace table name will be the name specified on the TRCTBL parameter. «

***OFF:** Collection of trace records stops. A spooled » file named QPCSMPRT is generated by the TRCINT command and contains the collected trace record data. TRCCNN formats this data in a second spooled file named QSYSPRT. The user data for the QSYSPRT spooled file is 'TRCCNN'. The trace table is deleted after the spooled files are generated.

***END:** Collection of trace records stops and the trace table is deleted. No spooled output is generated. «

***FORMAT:** Formats trace data in a QPCSMPRT spooled file created by a previous invocation of TRCCNN or TRCINT. The formatted data is written to a spooled file named QSYSPRT. The user data for the QSYSPRT spooled file is 'TRCCNN'. Use the TRCTYPE parameter to specify which connection-related trace records to format. Use the JOB and SPLNBR parameters to identify which QPCSMPRT file to use.

Optional Parameters

TRCTYPE

If SET(*ON) is specified, identifies the types of trace records to start collecting. If SET(*FORMAT) is specified, identifies the types of collected trace records to format. Multiple trace types may be specified.

***IP:** Trace IP (internet protocol) data.

***SSL:** Trace SSL (Secure Sockets Layer) connection data.

» **TRCTBL**

Specifies the trace table to hold the collected trace data.

GEN:** The trace table name will be **QTRCCNNxxxxxx* where *xxxxxx* is the job number of the current job.

For example, if TRCCNN SET(*ON) and TRCTBL(*GEN) are specified and the job identifier of the current job is 016870/QSRV/QPADEV000D, the generated trace table name is **TRCCNN016870**.

trace-table-name: Specify the name of the trace table to be used. If SET(*ON) is specified and the name specified does not match an existing trace table, a new trace table by the specified name will be created. «

SIZE Specifies the size of the trace table. The amount of storage to be allocated can be specified in units of kilobytes (*KB) or megabytes (*MB). If the size is specified in kilobytes, the amount of storage allocated for the table will be rounded up to the nearest megabyte. Valid table size values range from one megabyte to 258048 megabytes.

Note: The amount of storage specified by this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). This storage space is not dynamically allocated as it is needed. This storage space will not be available for use by the system except to record trace-related information. Before specifying a large value on this parameter, the amount of free space in the system ASP should be checked. Use the Work with System Status (WRKSYSSTS) command to determine the amount of available free space in the system ASP. System performance degradation may result if the size of the free space in the system ASP is significantly reduced as a result of the value specified.

Element 1: Size

16000: The trace table size is 16000 kilobytes or 16000 megabytes, depending on the Unit of measure value.

table-size: Specify the size of the trace table in kilobytes or megabytes. Valid values range from 128 through 258048.

Element 2: Unit of measure

Specifies whether the value specified for the Size element should be treated as number of kilobytes or number of megabytes.

***KB:** The trace table size is specified in kilobytes.

***MB:** The trace table size is specified in megabytes.

Single Values

***MAX:** The trace table is set to the maximum size of 258048 megabytes.

***MIN:** The trace table is set to the minimum size of one megabyte.

TRCFULL
Specifies whether the trace records wrap (replace the oldest records with new records) or stop tracing when the trace table is full.

***WRAP:** When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing is stopped when the trace table is full of trace records.

CCSID
Specifies whether the extended binary-coded decimal interchange code (*EBCDIC- 37) or the American National Standard Code for Information Interchange (*ASCII- 819) character code or any other is used for the formated output.

The possible values are:

***EBCDIC:** The EBCDIC (37) character code is used.

***ASCII:** The ASCII (819) character code is used.

coded-character-set-identifier: Specify a CCSID value to be used when formatting the trace data.
«

FMTDTA
Specifies the number of bytes of traced data to be formatted.

***CALC:** The system determines the number of bytes of data to be formatted.

number-of-bytes: Specify the number of bytes of data to be formatted. The minimum number of bytes allowed is 72.

JOB Specifies the name or qualified name of the job that created the input QPCSMPRT spooled file (SPLNBR parameter). This parameter is valid only if SET(*FORMAT) is specified.

***:** The job that issued this command is the job that created the input QPCSMPRT spooled file.

job-name: Specify the name of the job that created the input QPCSMPRT spooled file.

user-name: Specify the user name that identifies the user profile under which the job was run that created the input QPCSMPRT spooled file.

job-number: Specify the system-assigned job number of the job that created the input QPCSMPRT spooled file.

SPLNBR
Specifies the file number of the QPCSMPRT spooled file from the job (JOB parameter) that created the spooled file. This parameter is valid only if SET(*FORMAT) is specified.

***LAST:** The highest-numbered spooled file named QPCSMPRT created by the specified job is used.

***ONLY:** Only one spooled file named QPCSMPRT was created by the specified job; therefore, the number of the spooled file is not necessary. If *ONLY is specified and more than one spooled file for the specified job is named QPCSMPRT, an error message is issued.

spooled-file-number: Specify the number of the QPCSMPRT spooled file created by the specified job.

» **JOBSYSNAME**
Specifies the name of the system where the job that create the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

The possible values are:

***ONLY:** There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file creation date and time.

***CURRENT:** The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and creation date and time is used.

job-system-name: Specify the name of the system where the job that created the spooled file ran.

CRTDATE
Specifies the date and time the spooled file was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met.

***ONLY:** There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

***LAST:** The spooled file with the latest creation date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used. .

Element 1: Date Spooled File was Created.

date: Specify the date the spooled file was created.

Element 2: Time Spooled File was Created. ***ONLY:** There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name, and spooled file creation date.

***LAST:** The spooled file with the latest creation time of the specified job name, user name, job number, spooled file name, spooled file number, job system name and spooled file creation date is used.

time: Specify the time the spooled file was created. «

TCPDTA

Specifies whether a subset of TCP/IP and/or Sockets data should be collected. Each parameter element is optional; if no element value is specified, no filtering of trace data is done for that element. For example, if *ARP is specified for element 1, only trace records where the ARP protocol is used are collected. If no value is specified for element 1, trace records using all TCP/IP protocols are collected.

Element 1: Protocol

Specify a TCP/IP protocol to be traced.

***TCP:** Enable trace for transmission control protocol.

***UDP:** Enable trace for user datagram protocol.

***ICMP:** Enable trace for internet control message protocol.

***IGMP:** Enable trace for internet group management protocol.

***ARP:** Enable trace for address resolution protocol. This would only apply for TCP/IP.

***ICMP6:** Enable trace for internet control message protocol version 6.

Element 2: Local Ports

Specify one or two local port numbers for which trace data is collected.

Element 3: Remote Ports

Specify one or two remote port numbers for which trace data is collected

Element 4: Local IP Address

Specify a local internet protocol address in the form *nnn.nnn.nnn.nnn*, where *nnn* is a number between 1 and 255.

Element 5: Remote IP Address

Specify a remote internet protocol address in the form *nnn.nnn.nnn.nnn*, where *nnn* is a number between 1 and 255.

Element 6: Line Description Name

Specify the name of a line description for which TCP/IP trace data is to be collected.

Examples for TRCCNN

Example 1: Starting SSL Traces

```
TRCCNN  SET(*ON)  TRCTYPE(*SSL)
```

This command starts tracing for Secure Sockets Layer (SSL) connections.

Example 2: Starting IP Traces

```
TRCCNN  SET(*ON)  TRCTYPE(*IP)
```

This command starts tracing for connections at the internet protocol (IP) level.

Example 3: Stopping Traces and Clearing Trace Storage

```
TRCCNN  SET(*END)
```

This command stops all traces and deletes the trace table. No spooled output is generated.

Example 4: Printing Traces

```
TRCCNN  SET(*OFF)
```

This command stops all traces and generates a spooled file (QPCSMTRC) that contains the trace records collected by the TRCINT (Trace Internal) command, and a spooled file (QSYSPRT) that contains the formatted trace data.

Example 5: Formatting Trace Data from TRCINT Command

```
TRCINT SET(*ON) TRCTYPE(*SCK)
TRCINT SET(*OFF)
TRCCNN SET(*FORMAT) TRCTYPE(*SSL)
       JOB(*) SPLNBR(*LAST)
```

The TRCINT (Trace Internal) commands are used to start collecting trace records related to all usage of sockets, and to stop collecting trace records and create a spooled file named QPCSMPRT. The TRCCNN command will use the trace information in the last spooled file named QPCSMPRT for the current job, and format the trace records related to SSL (Secure Sockets Layer) in a spooled file named QSYSPRT.

» Example 6: Specifying a Trace Table

```
TRCCNN SET(*ON) TRCTYPE(*IP) TRCTBL(USER)
```

This command starts tracing for connections at the internet protocol (IP) level and stores the trace data in the USER trace table.

Example 7: Specifying a Trace Table Size in Megabytes

```
TRCCNN SET(*ON) TRCTYPE(*IP) SIZE(20000 *MB)
```

This command starts tracing for connections at the internet protocol (IP) level and stores the data in a 20000-megabyte trace table.

Example 8: Specifying a CCSID for Trace Data

```
TRCCNN SET(*OFF) CCSID(*ASCII)
```

This command stops all traces and generates a spooled file (QSYSPRT). ASCII (819) CCSID will be used when formatting the trace data.

Error messages for TRCCNN

No error messages. «

TRCCPIC (Trace CPI Communications) Command Description

TRCCPIC Command syntax diagram

Purpose

The Trace Common Programming Interface Communications (TRCCPIC) command controls tracing of all CPI Communications that occur in the job in which the command is entered. The command sets a trace on or off, and traces (1) CPI Communications calls issued by a program and (2) data that is sent and received.

As trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be directed to a spooled output file or a database physical file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCCPIC command, the job that is traced is the one specified on the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

Restrictions: (1) The record format of the database output file must match the record format of the IBM-supplied output file, QACM0TRC. (2) The user must have specific authority from the security officer to use this command.

Optional Parameters

SET Specifies whether a CPI Communications trace is started or ended.

***ON:** The trace is started. If the trace storage area becomes full, the action specified on the TRCFULL parameter is taken.

***OFF:** The trace is ended. No other trace information is recorded, and the current information is written to the spooled output file or a database file.

***END:** The trace ends. No other trace information is recorded and all current trace information is deleted. No output is generated.

MAXSTG
Specifies the maximum amount of storage (in kilobytes) used for the created trace records.

200: Up to 200KB of storage is used for trace records.

number-of-kilobytes: Specify the number of kilobytes of storage to use for trace records. Valid values range from 1 through 16000.

TRCFULL
Specifies the action taken when the maximum storage specified is full.

***WRAP:** When the trace storage area is full, new trace information is written over the old information, starting at the beginning of the storage area.

***STOPTRC:** When the trace storage area is full, no new trace information is saved.

DTALEN
Specifies the maximum length (in bytes) of user data that can be saved for each trace entry in the storage area. If the value specified is greater than the length of data received or sent across the communications line, only the actual data is traced. If the value specified is less than the data length received or sent, only the data length specified on this parameter is traced.

128: The maximum length of user data saved is 128 bytes.

number-of-bytes: Specify the maximum length of user data saved. Valid values range from 0 through 4096.

OUTPUT

Specifies whether the output from the command is printed with the job's spooled output or is 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 qualified name of the physical file to which the trace output is directed. If the file already exists, the system uses it. If the file does not exist, the system creates it. If the file is created, the text is "Output file for TRCCPIC."

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.

file-name: Specify the name of the physical file to which the trace output is directed.

OUTMBR

Specifies the name of the member in the physical file that receives the trace output. If the file is created by the system, a member is created with the name specified on this parameter. If the file exists but the member does not, a member with the specified name is created.

Element 1: Member to Receive Output

***FIRST:** The first member of the file specified on the OUTFILE parameter receives the trace output. If the file is created and *FIRST is specified, the name of the created member is the same as that of the file.

member-name: Specify the name of the member in the file that receives the trace output.

Element 2: Operation to Perform on Member

***REPLACE:** The new data replaces the existing data.

***ADD:** The new data is added to the file member at the end of the data already in the member.

Examples for TRCCPIC

Example 1: Starting Trace Operation

```
TRCCPIC  MAXSTG(350)  DTALEN(256)
```

This command traces the CPI Communications calls of the current job. The trace file contains 350KB of storage and wraps to the beginning if that amount of storage is filled with trace records. In addition, this command traces up to 256 bytes of user data on each input/output operation.

Example 2: Stopping Trace Operation

```
TRCCPIC  SET(*OFF)  OUTPUT(*OUTFILE)
  OUTFILE(TRACELIB/CPICTRACE)
  OUTMBR(TRACEMBR)
```

This command stops the trace and directs the output to the database file CPICTRACE in library TRACELIB. The output is directed to the member TRACEMBR.

Error messages for TRCCPIC

***ESCAPE Messages**

CPF2C90
Maximum storage specified too small.

CPF2C94
Error occurred during OUTFILE processing. Trace stopped.

CPF3B30
No CPI-Communications calls were run. Trace ended.

CPF3B31
Job is already being serviced or traced.

CPF3B32
Trace already off.

CPF3B33
Unexpected Trace CPI Communications error occurred.

CPF3B34
Cannot deactivate trace, trace started from another job.

CPF3548
Serviced job completed running.

CPF3936
Job being serviced ended before trace started.

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

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

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

TRCICF (Trace ICF) Command Description

TRCICF Command syntax diagram

Purpose

The Trace Intersystems Communications Functions (TRCICF) command controls tracing of all ICF I/O functions that occur in the job in which the command is entered. The command sets a trace on or off, and traces (1) ICF operations and functions issued by a program and (2) data that is sent and received.

As trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be directed to a spooled output file or a database physical file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCICF command, the job that is traced is the one specified on the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

Restrictions:

1. The record format of the database output file must match the record format of the IBM-supplied output file, QAIFTRCF.
2. The user must have specific authority from the security officer to use this command.

Optional Parameters

SET Specifies whether an ICF trace is started or ended.

***ON:** The trace is started. If the trace storage area becomes full, the action specified on the TRCFULL parameter is taken.

***OFF:** The trace is ended. No other trace information is recorded, and the current information is written to the spooled output file or a database file.

***END:** The trace ends. All trace information is deleted. No output is generated.

MAXSTG

Specifies the maximum amount of storage (in kilobytes) used for the created trace records.

200: Up to 200KB of storage is used for trace records.

number-of-kilobytes: Specify the number of kilobytes of storage to use for trace records. Valid values range from 1 through 16000.

TRCFULL

Specifies the action taken when the maximum storage specified is full.

***WRAP:** When the trace storage area is full, new trace information is written over old information, starting at the beginning of the storage area.

***STOPTRC:** When the trace storage area is full, no new trace information is saved.

DTALEN

Specifies the maximum length (in bytes) of user data that can be saved for each trace entry in the storage area. If the value specified is greater than the length of data received or sent across the communications line, only the actual data is traced. If the value specified is less than the data length received or sent, only the data length specified on this parameter is traced.

128: The maximum length of user data saved is 128 bytes.

number-of-bytes: Specify the maximum length of user data saved. Valid values range from 0 through 4096.

OUTPUT

Specifies whether the output from the command is printed with the job's spooled output or is 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 qualified name of the physical file to which the trace output is directed. If the file already exists, the system uses it. If the file does not exist, the system creates it. If the file is created, the text is “OUTFILE created by TRCICF command.” *EXCLUDE authority is assigned to users with no specific authority.

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.

file-name: Specify the name of the physical file to which the trace output is directed.

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.

Examples for TRCICF

Example 1: Starting Trace Operation

```
TRCICF  MAXSTG(350)  DTALEN(256)
```

This command traces the ICF input/output operations of the current job. The trace file contains 350K of storage and wraps to the beginning if that amount of storage is filled with trace records. In addition, this command traces up to 256 bytes of user data on each input/output operation.

Example 2: Stopping Trace Operation

```
TRCICF  SET(*OFF)  OUTPUT(*OUTFILE)
  OUTFILE(TRACELIB/ICFTRACE)
  OUTMBR(TRACEMBR)
```

This command stops the trace and directs the output to the database file ICFTRACE in library TRACELIB. The output is directed to the member TRACEMBR.

Error messages for TRCICF

*ESCAPE Messages

CPF2C90

Maximum storage specified too small.

CPF2C93

No trace records logged.

CPF2C94

Error occurred during OUTFILE processing. Trace stopped.

CPF2C95

Trace already active.

CPF2C96
Trace already off.

CPF3B34
Cannot deactivate trace, trace started from another job.

CPF3205
File not created.

CPF3501
Job is already being serviced, traced, or debugged.

CPF3530
Conflicting entries in index QSERVICE.

CPF3548
Serviced job completed running.

CPF3925
Cannot open file &1.

CPF3936
Job being serviced ended before trace started.

CPF3950
Error message &2 received for file &1. Request ended.

CPF3951
File &1 cannot be overridden by file name &2.

CPF3969
Error during close of file &1. Output may not be complete.

CPF5004
Printer overflow line detected for file &1.

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

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

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

TRCINT (Trace Internal) Command Description

TRCINT Command syntax diagram

Purpose

The Trace Internal (TRCINT) command is the command interface to the Trace Licensed Internal Code service tool and is used for problem analysis. Specific types of traces are started and stopped by using this command.

While previously started internal traces are being performed, additional internal traces can be started through this command. The output created by the trace is placed in a trace table. The records from the trace table can be written to a spooled printer file, or to tape or optical media.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.

2. 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.
3. The following user profiles have private authorities to use the command:
   - QSRV

Required Parameter

SET Specifies whether internal tracing is started, stopped, ended, held, or saved. Also, the user can specify whether the trace table size is changed.

***ON:** The collection of internal trace records is started for the trace types specified in the TRCTYPE parameter. If the trace table already contains trace records, the new trace records are added to the table. If the table is full, the action specified in the TRCFULL parameter is taken.

If a trace table name other than *SYSDFT is specified for the TRCTBL parameter and the table does not exist, it will be automatically created.

***OFF:** Collection of internal trace records requested through previous TRCINT commands stops, and the records are written to the spooled printer file QPCSMPRT.

***END:** Internal tracing ends and the trace records are deleted. No spooled output is generated.

If a trace table name other than *SYSDFT is specified for the TRCTBL parameter, it will be automatically deleted.

***HOLD:** Internal traces are stopped, and the collected trace records are held in the trace table. Held records can be printed later if another TRCINT command is entered that specifies SET(*OFF), or they can be put on tape or optical media if SET(*SAVE) is specified.

***SAVE:** Internal traces are stopped, and the trace records are written to the tape or optical device specified by the OUTDEV parameter. The diskettes on which the records are written must be initialized in the basic exchange format (*DATA or *DATA2).

***SIZE:** The size of the trace table is changed. The new size is specified in the SIZE parameter.

Optional Parameters

TRCTBL
Specifies the trace table to hold the collected trace data.

***SYSDFT:** The system default trace table is used.

trace-table-name: Specify the name of the trace table to be used. If SET(*ON) is specified and the name specified does not match an existing trace table, a new trace table by the specified name will be created.

SIZE Specifies the size of the trace table. This parameter can be specified only when SET(*SIZE) is specified, or if SET(*ON) is specified and tracing is not currently active for the trace table specified (TRCTBL parameter).

Note: The storage indicated on this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). This storage is not dynamically allocated as it is needed. This storage will not be available for use by the system except to record trace-related information. Before specifying a large value on this parameter, the amount of free space in the system ASP should be checked. Use the Work with System Status (WRKSYSSTS) command to determine the amount of available free space in the system ASP. System performance degradation may result if the size of the free space in the system ASP is significantly reduced as a result of the value specified.

» Element 1: Size «

***NOCHG:** The trace table size is not changed. If a new trace table is specified (TRCTBL parameter), a default size of 128 kilobytes will be used.

***MAX:** The trace table is set to the maximum size of » 258048 megabytes « .

***MIN:** The trace table is set to the minimum size of 128 kilobytes.

table-size: Specify the size of the trace table in » kilobytes or megabytes « . Valid values range from » 1 « through 998000.

» Element 2: Unit of Measure

Specify whether the table-size value specified for the Size element should be treated as number of kilobytes or number of megabytes.

***KB:** The trace table size is specified in kilobytes. The valid range is 128 through 998000.

***MB:** The trace table size is specified in megabytes. The valid range is 1 through 258048. «

TRCFULL

Specifies whether the trace records wrap (replace the oldest records with new records) or tracing stops when the trace table is full. This parameter can be specified only when SET(*ON) is specified.

***NOCHG:** The trace table full action is not changed. If a new trace table is specified (TRCTBL parameter), the default action is for trace records to wrap when the trace table becomes full.

***WRAP:** When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing is stopped when the trace table is full of trace records.

TRCTYPE

Specifies the types of traces to start. The two groups of trace types are:

1. Component data trace codes: Active procedures are traced within the system.
2. General trace codes: The iSeries 400 instruction supervisor linkage, multiprogramming, transaction, or task and thread performance functions are traced.

This parameter can be specified only if SET(*ON) is specified. If a value other than *ON is specified on the SET parameter, TRCTYPE is ignored. Each trace type is identified by a special value or six-digit code. If a numeric value is specified, all 6 digits must be specified. Specify up to 50 trace types from those listed in Table 1 (page 58).

For a complete list of trace codes and special values, position the cursor on the field for the **Trace type** prompt (TRCTYPE parameter), and press F4.

Table 1. Trace Code Table

	Trace Code	
Type of Trace	**Component or General**	**Special Value**
MI instruction supervisor linkage (SVL)	030000	*SVL
Multi-programming level (MPL)	040000	*MPL
Transaction	080000	*TNS
Task/thread performance	090000	*TTPERF
Activation/Call	014400	*ACTCALL
MPTN - APPC over TCP/IP MPTN	014203	*APPCOVRTCP
APPC presentation services	014301	*APPCPS
APPN - all	012506	*APPNALL
APPN control point management	012501	*APPNCPM
APPN control point presentation services	012504	*APPNCPPS
APPN directory services	012502	*APPNDS
APPN location management	012505	*APPNLM
APPN management services transport	012507	*APPNMST
APPN topology and routing services	012503	*APPNTRS
Authority management	010900	*AUTMGT
Auxiliary storage management - detailed	011104	*AUXSTGALL
Auxiliary storage management	011101	*AUXSTGMGT
Byte string space management	012600	*BSSMGT
Common class input/output management (CCIOM)	011900	*CCIOM
Cluster engine	016402	*CLUE
Communications access method	015900	*CMNACCMTH
Commit management	011700	*CMTMGT
Communications trace service function	012300	*CMNTRC
Common functions	011200	*COMMON
Cryptographic services	013600	*CRPSRV
Cluster - all	016400	*CSTALL
Cluster communications	016401	*CSTCMN
Context management	011000	*CTXMGT
Debugger interpreter	014500	*DBGINT
Database management (events for all database files are traced)	010400	*DBMGT
Dependent LU Requestor (DLUR) Communication	015400	*DLUR
Display station pass-through	010804	*DSPPASTHR
Environmental recording, editing and printing (EREP)	012200	*EREP
Error log	012100	*ERRLOG
Event management	010600	*EVTMGT
Exception management	010200	*EXCMGT
» Fast Response Cache Accelerator	016600	*FRCA «
Hardware resources	014700	*HDWRSC
Heap management	013400	*HEAPMGT
Independent index management	011400	*IDXMGT
Integrated File System (IFS)	014800	*IFS
Inter-process communications facility (IPCF)	012000	*IPCF
Communications answer management (ISDN)	012700	*ISDN
Journal management	011600	*JRNMGT
Link test service function	012400	*LNKTST

Load/dump (save/restore)	010801	*LODDMP
Machine observation	011300	*MCHOBS
Machine Interface (MI) Transformer	015100	*MITFMALL
MI Transformer - expression evaluation	015101	*MITFMEVAL
MI Transformer - heap operations	015102	*MITFMHEAP
MI Transformer - interpreter instructions	015103	*MITFMINT
MI Transformer - MI instructions	015104	*MITFMMI
MI Transformer - storage management operations	015105	*MITFMSTG
Module management	013100	*MODMGT
Modula-2 run time support	012800	*MOD2
Machine services control point	010802	*MSCP
Main storage management - calls	011102	*MSMCALL
Main storage management - details	011103	*MSMDTL
AS/400 Advanced 36 - all	014600	*M36ALL
AS/400 Advanced 36 - Async communications	014605	*M36ASC
AS/400 Advanced 36 - BSC communications	014604	*M36BSC
AS/400 Advanced 36 - supervisor (CSP)	014611	*M36CSP
AS/400 Advanced 36 - diskette	014607	*M36DKT
AS/400 Advanced 36 - disk	014610	*M36DSK
AS/400 Advanced 36 - internal LAN communications	014612	*M36ILAN
AS/400 Advanced 36 - printer	014609	*M36PRT
AS/400 Advanced 36 - SDLC communications	014601	*M36SDLC
AS/400 Advanced 36 - tape	014606	*M36TAP
AS/400 Advanced 36 - Token Ring communications	014602	*M36TRN
AS/400 Advanced 36 - workstation	014608	*M36WS
AS/400 Advanced 36 - X.25 communications	014603	*M36X25
NetBIOS on TCP/IP	015700	*NTBTCP
OptiConnect	015500	*OPC
Private address space environment	016100	*PASE
Performance collection services	016200	*PFRCOLSRV
Program binder	013200	*PGMBND
Program management	010300	*PGMMGT
Portability utilities	015200	*PORTUTIL
Process management	010500	*PRCMGT
Process table	015300	*PRCTBL
» Pseudo terminal component	016500	*PSEUDOTERM «
Power management	012900	*PWRMGT
Queue management	010700	*QMGT
Queue space management	013300	*QSMGT
Recovery management	013500	*RCYMGT
Remote support	016300	*RMTSPT
Resource management	010100	*RSCMGT
Sockets	014000	*SCK
Sockets Asynchronous and Overlapped Input/Output APIs	014002	*SCKASCIO
Sockets Network APIs	014004	*SCKNET
Other Socket APIs	014007	*SCKOTHER
MPTN AF_INET Sockets over SNA MPTN	014201	*SCKOVRMPTN
MPTN AF_INET Sockets over SNA PEC	014202	*SCKOVRPEC
Sockets Berkeley Resolver APIs	014005	*SCKRSLV
Sockets Select API	014003	*SCKSEL
Secure Sockets Layer (SSL) APIs	014006	*SCKSSL
Sockets Standard Input/Output APIs	014001	*SCKSTDIO
Signals	015600	*SIG
SMB server	015800	*SMBSVR
Space object management	011500	*SPCOBJMGT

Source/sink (device support) management	010803	*SRCSINK
Storage management - all	011105	*STGMGTALL
Streams	013900	*STM
Synchronization management	013700	*SYNCMGT
MI system call	015200	*SYSCALL
System journal management	014100	*SYSJRNMGT
Transmission Control Protocol/Internet Protocol (TCP/IP)	013800	*TCPIP
Transport Layer Interface	016000	*TLI
» Transaction Management	016800	*TRXMGT «
» Virtual I/O	016700	*VRTIO «
Virtual terminal management	013000	*VRTDEVMGT

» **JOB**

Specifies the name of a job (job-number/user-name/job-name) which is used to determine whether the trace record is collected. Only trace records which are generated in the specified job(s) are collected. A list of up to ten qualified job names can be specified. The trace record will be collected if it was generated from a job that matches any of the qualified job name values.

Single Value

***NOCHG:** If any qualified job names had been specified for the JOB parameter on a previous TRCINT command for an active trace, the job name filtering information is not changed. If no value was specified on a previous TRCINT command for an active trace, *NOCHG will behave the same as *ALL/*ALL/*ALL for the JOB parameter.

Job Name

***ALL:** All trace records generated by the defined trace are collected, regardless of what job name the trace record was generated from.

generic-job-name:* Specify the generic name of the job from which trace records are to be collected. A generic name is a character string of one or more characters followed by and 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-name: Specify the name of the job from which trace records are to be collected.

Job User Name

***ALL:** All trace records generated by the defined trace are collected, regardless of what job user name the trace record was generated from.

generic-user-name:* Specify the generic user name of the job from which trace records are to be collected. A generic name is character string of one or more characters followed by and asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic user name specifies all jobs with user names that begin with the generic prefix.

user-name: Specify the name of the user of the job from which trace records are to be collected.

Job Number

***ALL:** All trace records generated by the defined trace are collected, regardless of what job number the trace record was generated from. *ALL for the job number is considered to be a generic job specification because it will trace all jobs that meet the job name and job user name qualifiers that you specified.

job-number: Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name or generic user name is specified.

SLTTHD

Specifies a thread identifier number which is used to determine whether the trace record is collected. Only trace records which are generated in the specified thread(s) are collected. There can be only one job that has thread IDs associated with it. It must be the first qualified job name specified for the JOB parameter, and the job must be active.

***NOCHG:** If any thread identifiers had been specified for the SLTTHD parameter on a previous TRCINT command for an active trace, the thread ID filtering information is not changed. If no value was specified on a previous TRCINT command for an active trace, *NOCHG will behave the same as *ALL for this parameter.

***ALL:** All trace records generated by the defined trace are collected, regardless of what thread ID the trace record was generated from.

***SELECT:** A list of thread identifiers is shown from which you can select up to twenty threads. Trace records from any of the selected thread identifiers are to be collected. *SELECT is only valid if the TRCINT command is run in an interactive job.

thread-identifier: Specify the identifier of the thread from which trace records are to be collected. Up to twenty thread identifiers can be specified.

SVRTYPE

Specifies the server type attribute for a job or task which is used to determine whether the trace record is collected. Only trace records which are generated in a job or task with the specified server type are collected.

***NOCHG:** If a value was specified for the server type on a previous TRCINT command for an active trace, the value is not changed. If no value was specified on a previous TRCINT command, *NOCHG will behave the same as *ALL for this parameter.

***ALL:** All trace records generated by the defined trace are collected, regardless of the server type attribute of the job or task the trace record was generated from.

generic-server-type:* Specifies the generic server type for which trace records are to be collected. A generic name is character string of one or more characters followed by and asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic server type specifies all jobs with a server type that begins with the generic prefix.

server-type: Specify the server type for which trace records are to be collected. A list of up to five server types can be specified.

TASK Specifies the name of a Licensed Internal Code (LIC) task which is used to determine whether the trace record is collected. Only trace records which are generated from the specified LIC task are collected.

***NOCHG:** If a value was specified for the task name (TASK parameter) on a previous TRCINT command for an active trace, the value is not changed. If no value was specified on a previous TRCINT command, *NOCHG will behave the same as *ALL for the task name.

***ALL:** All trace records generated by the defined trace are collected, regardless of what LIC task the trace record was generated from.

generic-task-name:* Specify the generic name of the LIC tasks for which trace records are to be collected. A generic name is a character string of one or more characters followed by and asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic task name specifies all tasks with task names that begin with the generic prefix.

task-name: Specify the name of the LIC task for which trace records are to be collected. Up to ten LIC task names can be specified.

TASKNBR

Specifies the task number of a Licensed Internal Code (LIC) task which is used to determine whether the trace record is collected. Only trace records which are generated in the specified LIC task are collected.

***NOCHG:** If a value was specified for the task number on a previous TRCINT command for an active trace, the value is not changed. If no value was specified on a previous TRCINT command, *NOCHG will behave the same as *ALL for this parameter.

***ALL:** All trace records generated by the defined trace are collected, regardless of the task number of the LIC task the trace record was generated from.

task-number: Specify a LIC task number for which trace records are to be collected. A list of up to ten task numbers can be specified. «

SLTTRCPNT

Specifies a list of up to five individual trace points or trace point ranges whose trace records are to be included. Trace records for trace points not specified on SLTTRCPNT will not be collected. If SLTTRCPNT is specified for a trace table that is currently active, the specified trace points will be added to the set of trace points for which trace records are being collected.

Note: This parameter and the OMTTRCPNT parameter are mutually exclusive. SLTTRCPNT cannot be specified for an active trace table that is using OMTTRCPNT to exclude specific trace points.

***NOCHG:** The list of trace points for which trace records are being collected does not change.

trace-point-qualifier: Specify a single trace point qualifier or a range of trace point qualifiers whose trace records are to be included. Up to five individual trace point qualifiers or trace point qualifier ranges may be specified.

OMTTRCPNT

Specifies a list of up to five individual trace points or trace point ranges whose trace records are to be excluded. Trace records for all trace points not specified on OMTTRCPNT will be collected. If OMTTRCPNT is specified for a trace table that is currently active, the specified trace points will be added to the set of trace points for which trace records are not being collected.

Note: This parameter and the SLTTRCPNT parameter are mutually exclusive. OMTTRCPNT cannot be specified for an active trace table that is using SLTTRCPNT to include only specific trace points.

***NOCHG:** The list of trace points for which trace records are being excluded does not change.

trace-point-qualifier: Specify a single trace point qualifier or a range of trace point qualifiers whose trace records are to be excluded. Up to five individual trace point qualifiers or trace point qualifier ranges may be specified.

STOPTRCPNT

Specifies one or more trace points which, if they are encountered, will cause collection of trace records to stop. The trace table records are not deleted and can later be written to a spooled file or an output device by invoking TRCINT with SET(*OFF) or SET(*SAVE).

Up to four trace points may be specified. Tracing will be stopped if any of the specified trace points match a trace record being added to the specified trace table.

A specified trace point can have either two parts (trace point type and trace point qualifier) or five parts (trace point type, trace point qualifier, trace point entry number, trace point entry offset, and trace point match value). A two-part condition will stop trace data collection if any trace record is collected for the specified trace point. A five-part condition will stop trace data collection only if the specified trace point match value exactly matches the data at the specified trace point entry offset.

Single Value

***NOCHG:** The list of stop trace points associated with the trace table does not change.

Element 1: Trace Point Type

Specify the two-character trace point type.

Element 2: Trace Point Qualifier

Specify the trace point qualifier number. Valid numbers are 0 to 65535

Element 3: Trace Point Entry Number

Specify the trace point entry number.

Element 4: Trace Point Entry Offset

Specify the offset (in hexadecimal) in the trace point entry.

Element 5: Trace Point Match Value

Specify the match value to be compared to the trace record data. The match value may be specified in character or hexadecimal. Character strings will be converted to the equivalent hexadecimal strings.

JOBTYPE

Specifies the types of jobs for which trace data is being collected for use in the batch job trace report. A maximum of 11 of the following job types can be traced, or one of the single values *NONE or *ALL can be specified. This parameter can be specified only if TRCTYPE(*MPL) or TRCTYPE(040000) is specified.

Note: The value *DFT includes the values *ASJ, *BCH, *EVK, *MRT, *PDJ, *PJ and *BCI. The value *BCH includes the values *EVK, *MRT, *PDJ, *PJ, and *BCI.

*DFT: Batch and autostart job types are traced.

***ASJ:** Autostart job types are traced.

***BCH:** Batch job types are traced.

***EVK:** Jobs started (evoked) by a procedure start request are traced.

***INT:** Interactive jobs are traced.

***MRT:** Multiple requester terminal job types are traced.

***RDR:** Reader job types are traced.

***SBS:** Subsystem monitor job types are traced.

***SYS:** System job types are traced.

***WTR:** Writer job types are traced.

***PDJ:** Print driver job types are traced.

***PJ:** Prestart job types are traced.

***BCI:** Batch Immediate job types are traced.

***NONE:** No job types are traced.

***ALL:** All of the job types listed above are traced.

JOBTRCITV

Specifies the time (in CPU seconds) between each collection of the job trace data. This parameter can be specified only if TRCTYPE(*MPL) or TRCTYPE(040000) is specified.

0.5: A time slice interval value of 0.5 CPU seconds is used.

number-of-seconds: Specify a time slice interval value ranging from 0.1 through 9.9 CPU seconds.

TCPDTA

Specifies whether a subset of TCP/IP and/or Sockets trace data should be collected. This parameter can be specified only if TRCTYPE(*TCPIP) or TRCTYPE(013800) is specified or if one or more of the sockets trace types is specified (*SCK, *SCKSTDIO, *SCKASCIO, *SCKSEL, *SCKNET, *SCKRSLV, *SCKSSL, *SCKOTHER, or 014000, 014001, 014002, 014003, 014004, 014005, 014006, 014007). Each parameter element is optional; if no element value is specified, no filtering of TCP/IP or Sockets trace data is done for that element. For example, if *UDP is specified for element 1, only trace records where the UDP protocol is used are collected. If no value is specified for element 1, trace records using all TCP/IP protocols are collected.

If no values are specified for any element of TCPDTA and tracing of TCP/IP or Sockets data was not already active, no filtering of TCP/IP or Sockets trace data is done. If tracing of TCP/IP or Sockets data was already active and no TCPDTA values are specified, previous data filtering values will remain in effect.

Element 1: Protocol

Specify a TCP/IP protocol to be traced.

***TCP:** Enable trace for transmission control protocol.

***UDP:** Enable trace for user datagram protocol.

***ICMP:** Enable trace for internet control message protocol.

***IGMP:** Enable trace for internet group management protocol.

***ARP:** Enable trace for address resolution protocol. This will only apply for TCP/IP.

» ***ICMP6:** Enable trace for internet control message protocol version 6. «

Element 2: Local Ports

Specify one or two local port numbers for which trace data is collected.

Element 3: Remote Ports

Specify one or two remote port numbers for which trace data is collected.

Element 4: Local IP Address

» Specify a local internet protocol address. «

Element 5: Remote IP Address

» Specify a remote internet protocol address. «

Element 6: Line Description Name

Specify the name of a line description for which TCP/IP trace data is to be collected.

Element 7: Line Type

Specify whether the collection of trace information should be restricted to the specified line type.

***PPP:** The collection of trace information is restricted to Point-to-point lines.

***OPC:** The collection of trace information is restricted to Opticonnect.

SCKDTA

Specifies whether a subset of Sockets trace data should be collected. This parameter can be specified only if one or more of the socket trace types is specified (*SCK, *SCKSTDIO, *SCKASCIO, *SCKSEL, *SCKNET, *SCKRSLV, *SCKSSL, *SCKOTHER or 014000, 014001, 014002, 014003, 014004, 014005, 014006, 014007). Each parameter element is optional; if no element value is specified, no filtering of Sockets trace data is done for that element. For example, if *INET is specified for element 1, only trace records where the AF_INET address family is used are collected. If no value is specified for element 1, trace records using all socket address families are collected.

If no values are specified for any element of SCKDTA and tracing of Sockets data was not already active, no filtering of Sockets trace data is done. If tracing of Sockets data was already active and no SCKDTA values are specified, previous data filtering values will remain in effect.

The subset values specified on the SCKDTA parameter are used in combination with any subset values specified on the TCPDTA parameter to generate the complete subsetting criteria.

Element 1: Address Family

Specify a sockets address family for which trace data is collected.

***INET:** Enable trace for AF_INET address family.

***UNIX:** Enable trace for AF_UNIX and AF_UNIX_CCSID address families.

***TELEPHONY:** Enable trace for AF_TELEPHONY address family.

***NETBIOS:** Enable trace for AF_NETBIOS address family.

Element 2: Socket Type

Specify a socket type for which trace data is collected.

***STREAM:** Enable trace for SOCK_STREAM (full-duplex stream) socket type.

***DGRAM:** Enable trace for SOCK_DGRAM (datagram) socket type.

***RAW:** Enable trace for SOCK_RAW (direct to network protocol) socket type.

***SEQPACKET:** Enable trace for SOCK_SEQPACKET (full-duplex sequenced packet) socket type.

Element 3: Descriptor Number

Specify one or two socket descriptor numbers for which trace data is collected.

Element 4: Socket Option

Specify a socket option for which trace data is collected.

***SODEBUG:** Applications with the SO_DEBUG Socket Option set on will have trace data collected.

DEV Specifies the names of the devices for which the associated internal events are traced. This parameter can be specified only if TRCTYPE(*SRCSINK) or TRCTYPE(010803) is specified. Up to 16 device names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the DEV parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(*ON) multiple times for the same trace table

- specifying *ALLDEV on the CTL parameter
- specifying *ALLCTL on the LIN parameter

***NONE:** No devices are traced by this command.

device-name: Specify the name of the device for which the internal trace is started. The device name must be the same as the name specified in the associated device description.

CTL Specifies the names of the controllers for which the associated internal events are traced. This parameter is valid only if TRCTYPE(*SRCSINK) or TRCTYPE(010803) is specified. Up to 16 controller names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the CTL parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(*ON) multiple times for the same trace table
- specifying *ALLDEV on the CTL parameter
- specifying *ALLCTL on the LIN parameter

Single Value

***NONE:** No controllers are traced by this command.

Element 1: Controller Name

controller-name: Specify the name of the controller for which the internal trace is started. The controller name must be the same as the name specified in the associated controller description.

Element 2: Tracing of Attached Devices

Specifies if the devices on a controller are traced.

***NODEV:** No attached devices for the specified controller are traced.

***ALLDEV:** All attached devices for the specified controller are traced. The attached devices do not count toward the maximum of 16 source/sink objects that can be named on the DEV, CTL, LIN, NWI and NWS parameters. However, the attached devices do count toward the maximum of 256 source/sink objects that can be traced in a single trace table.

LIN Specifies the names of the lines for which the associated internal events are traced. This parameter is valid only if TRCTYPE(*SRCSINK) or TRCTYPE(010803) is specified. Up to 16 line names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the LIN parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(*ON) multiple times for the same trace table
- specifying *ALLDEV on the CTL parameter
- specifying *ALLCTL on the LIN parameter

Single Value

***NONE:** No lines are traced by this command.

Element 1: Line Name

line-name: Specify the name of the line for which the internal trace is started. The line name must be the same as the name specified in the associated line description.

Element 2: Tracing of Attached Controllers

Specifies if the controllers on a line are traced.

***NOCTL:** No attached controllers for the specified line are traced.

***ALLCTL:** All attached controllers for the specified line are traced. The attached controllers do not count toward the maximum of 16 source/sink objects that can be named on the DEV, CTL, LIN, NWI and NWS parameters. However, the attached controllers do count toward the maximum of 256 source/sink objects that can be traced in a single trace table.

NWI Specifies the names of the network interfaces for which the associated internal events are traced. This parameter can be specified only if TRCTYPE(*SRCSINK) or TRCTYPE(010803) is specified. Up to 16 network interface names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the NWI parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(*ON) multiple times for the same trace table
- specifying *ALLDEV on the CTL parameter
- specifying *ALLCTL on the LIN parameter

***NONE:** No network interfaces are traced by this command.

network-interface-name: Specify the name of the network interface for which the internal trace is started. The network interface name must be the same as the name specified in the associated network interface description.

NWS Specifies the names of the network servers for which the associated internal events are traced. This parameter can be specified only if TRCTYPE(*SRCSINK) or TRCTYPE(010803) is specified. Up to 16 network server names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the NWS parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(*ON) multiple times for the same trace table
- specifying *ALLDEV on the CTL parameter
- specifying *ALLCTL on the LIN parameter

***NONE:** No network servers are traced by this command.

network-server-name: Specify the name the network server for which the internal trace is started. The network server name must be the same as the name specified in the associated network server description.

RSRCNAME

Specifies the name of the hardware resource for which the associated internal events are traced. This parameter can be specified only if » TRCTYPE(014700) or TRCTYPE(011900) is specified. « A maximum of » ten « hardware resource names may be specified.

***NONE:** No hardware resource is traced by this command.

hardware-resource-name: Specify the name of a hardware resource for which the internal trace is started.

OUTDEV

Specifies the device description name of the tape or optical device on which the held trace records are written. This parameter must be specified if SET(*SAVE) is specified; otherwise it is ignored.

TASKINF

Specifies whether information for all licensed internal code (LIC) tasks is written to a spooled file or output device. This parameter can be specified if SET(*OFF) or SET(*SAVE) is specified.

***TRCREF:** Write information only for LIC tasks that were referenced by trace records in the specified trace table.

***ALL:** Write information for all LIC tasks that were in existence while the trace was active.

Examples for TRCINT

Example 1: Starting Component Data Traces and Call Traces

```
TRCINT  SET(*ON)  TRCTYPE(010100 010400 050500 051200)
```

This command starts component data traces and call traces for resource management and database. Database operations associated with database files are used to collect component data trace records.

Example 2: Tracing Lines and Controllers

```
TRCINT  SET(*ON)  TRCTYPE(*SRCSINK) TRCTBL(*SYSDFT)
        DEV(WS1 WS2 WS3)
        CTL((C1) (C2)) LIN((L1) (L2))
```

This command starts component data traces for source/sink management (device support) operations involving the devices WS1, WS2, and WS3, lines L1 and L2, and controllers C1 and C2.

Example 3: Stopping Traces and Clearing Trace Table

```
TRCINT  SET(*END)  TRCTBL(*SYSDFT)
```

This command stops all traces and deletes the trace records from the system default trace table.

Example 4: Tracing Communications Trace Service Function

```
TRCINT  SET(*ON)  TRCTYPE(*CMNTRC)
```

This command starts component data traces for the communications trace service function.

» Example 5: Using Job Filtering Capability

```
TRCINT  SET(*ON) TRCTBL(MYFTPTRACE) TRCTYPE(*TCPIP)
   JOB(QTCP/QTFTP*)
```

This command starts a TCP/IP trace and will only collect trace records for trace points collected in jobs with user name QTCP and job names that begin with the prefix QTFTP. Trace records will be stored in trace table MYFTPTRACE. «

Error messages for TRCINT

*ESCAPE Messages

CPD3683
TRCFULL parameter only valid with SET(*ON).

CPD3685
SLTTRCPNT or OMTTRCPNT parameters are mutually exclusive.

CPD3686
TCPDTA only valid if TRCTYPE(*TCPIP) is specified.

CPD3688
Job types (JOBTYPE) parameter only valid with TRCTYPE(*MPL).

CPD3689
Job trace interval (JOBTRCITV) parameter only valid with TRCTYPE(*MPL)

CPD368A
Cannot change trace point selection criteria for active trace table.

CPD36C0
OUTDEV parameter only valid with SET(*SAVE).

CPD36C1
SIZE parameter only valid with SET(*ON) or SET(*SIZE).

CPD36CD
TASKINF parameter only valid with SET(*OFF) or SET(*SAVE).

» **CPD3983**
Range of parameter SIZE not valid.

CPD3990
User number qualifier not valid.

CPD3991
Job, thread identifier or task not active.«

CPF3515
Too many trace requests or objects.

CPF3516
Trace table is full.

» **CPF3517**
Cannot specify *SELECT for the thread ID to include.«

CPF3518
End time and date earlier than start time and date.

CPF3659
Total of specified CTL, DEV, LIN, NWI, and NWS greater than allowed.

CPF3679
Service function returned completion code &1 qualifier &2.

CPF368A
Trace table size not changed.

CPF3683
Error occurred trying to open printer file.

CPF3684
Error occurred while trying to close a print file.

CPF3685
Error occurred while data being put to print file.

CPF3686
Service function ended with error message.

CPF3687
Error occurred while trying to open file.

CPF3688
Error occurred while tape or diskette file being closed.

CPF3689
Error occurred while writing data to tape or diskette.

CPF3692
Error occurred while trying to write data to tape or diskette.

CPF3693
Service function ended because error occurred.

CPF3694
Cannot start service function.

CPF3695
No trace tables exist.

CPF3696
No traces recorded.

CPF3697
Trace type parameter value missing.

CPF7A11
Trace table &1 not found.

CPF7A13
Trace table cannot be created.

CPF7A15
Trace buffer must be cleared.

CPF7A17
Trace already being active.

CPF7A1A
Cannot change trace point selection criteria for active trace table.

CPF7A1C
IP address not valid.

TRCJOB (Trace Job) Command Description

TRCJOB Command syntax diagram

Purpose

The Trace Job (TRCJOB) command controls traces of original program model (OPM) programs and Integrated Language Environment (ILE) procedure calls and returns that occur in the current job or in the job being serviced as a result of the Start Service Job (STRSRVJOB) command directed to that job. This command, which sets a trace on or off, can trace module flow, OS/400 system data acquisition (including CL command traces), or both.

As the trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be written to a spooled printer file, QPSRVTRC. The trace records can also be directed to a database output file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCJOB command, the job that is traced is the one identified by the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

Restrictions:

1. The record format of the database output file must match the record format of the IBM-supplied output file QATRCJOB.
2. The number of trace records processed between the start and end of the trace must not exceed one million.
3. This command is shipped with public *EXCLUDE authority.
4. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSRV
   - QSRVBAS
   - QSYSOPR
   - QRJE

Required Parameters

SET Specifies whether the collection of trace records starts or stops.

***ON:** The collection of trace records is started.

***OFF:** The collection of trace records is stopped, and the trace records are written to the spooled printer file.

***END:** The collection of trace records is stopped, and all existing trace records are deleted. No spooled printer file is created.

TRCTYPE
Specifies the type of trace data to be stored in a trace file.

***ALL:** All the trace data collected is stored in trace records. This includes tracing the flow of control and the trace data itself.

***FLOW:** The flow of control is traced when OPM programs and ILE procedures are called and when they return control.

***DATA:** The data that is provided at predefined trace points within the OS/400 system is stored in trace records. This includes trace records for the CL commands that have run.

MAXSTG
Specifies the maximum amount of storage in kilobytes (K) used for collected trace records.

4096: Up to 4096 K-bytes of storage is used.

maximum-K-bytes: Specify the maximum amount of storage, in K-bytes, used to store trace records (one K equals 1024 bytes).

Note: The maximum amount of storage, in kilobytes, is 16000.

TRCFULL

Specifies whether the trace records wrap (replace oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

***WRAP:** When the trace file is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing stops when the trace file is full of trace records.

EXITPGM

Specifies the qualified name of a user-written program that is given control just prior to the collection of each trace record.

Note: Items being traced can be missed when an exit program is used. Do not use an exit program if you do not want to risk losing a trace record.

***NONE:** No user-written program is called.

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

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

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

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

program-name: Specify the name of the user-written program called before each trace record is collected. This program can examine the trace record passed to it as a parameter, and alter the first two characters. If it replaces the first two characters in the trace record with blanks or binary zeros, the entire trace record is suppressed.

This program can also call any of the service dump commands and must return control when it completes its task. If the user-written program runs other OS/400 system commands, the output from those commands is associated with the job in which they are run. The dump and trace commands associate their output data either with the job that enters the dump and trace commands or with the job being serviced as the result of the STRSRVJOB command.

SLTPRC

Specifies which ILE procedure calls and returns are included in the trace.

***ALL:** All ILE procedure calls and returns are included in the trace.

***NONE:** No ILE procedure calls or returns are included in the trace.

Element 1: Program Name

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 names of a maximum of 50 bound programs and bound service programs for which all procedure calls and returns are included in the trace.

Element 2: Program Type

***PGM:** The specified program is a bound program.

***SRVPGM:** The specified program is a bound service program.

SLTTHD

Specifies a list of up to twenty threads whose calls and returns are included in the trace. Only trace records for the specified thread identifiers are included.

***ALL:** All threads calls and returns are included in the trace.

***SELECT:** A list of thread identifiers is shown from which the user can select up to twenty whose trace records are to be included.

thread-identifier: Specify the identifiers of up to twenty threads whose trace records are to be included.

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.

***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 physical database file to which the trace output is directed. If the output file already exists, the system tries to use it. Records replace existing data in the file member. If the output file does not exist, the system creates a database physical 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 the file is created, the text is "OUTFILE created by TRCJOB command," and *EXCLUDE authority is given to users with no specific authority.

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 to receive the trace output.

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

Examples for TRCJOB

Example 1: Tracing Flow of Control

```
TRCJOB  TRCTYPE(*FLOW)  MAXSTG(40)
```

This command traces the flow of the current job. Trace records are collected for each OPM program and ILE procedure call and return that occurs in the job. The trace file contains 40K of storage and wraps (oldest records are replaced by new records) if that amount of storage is filled with trace records.

Example 2: Stopping the Trace Operation

```
TRCJOB  SET(*OFF) OUTPUT(*OUTFILE) OUTFILE(QGPL/TRCJOB)
  OUTMBR(TRCDTA)
```

This command stops the trace and directs the output to the database file QGPL/TRCJOB. The output is directed to the member TRCDTA.

Example 3: Tracing Flow of Control - Selecting Specific ILE Procedures

```
TRCJOB  SET(*ON)  TRCTYPE(*FLOW)
  SLTPRC((MYLIB/MYPGM1 *PGM) (MYLIB/MYSRVPGM1 *SRVPGM))
```

This command traces the flow of the current job. Trace records are collected for all OPM program calls and returns and the ILE procedure calls and returns of bound program MYPGM1 and bound service program MYSRVPGM1.

Example 4: Tracing one thread

```
TRCJOB  SET(*ON)  SLTTHD(00000001)
```

This command traces only the specified thread on the job.

Error messages for TRCJOB

***ESCAPE Messages**

CPF2C94
Error occurred during OUTFILE processing. Trace stopped.

CPF2C95
Trace already active.

CPF2C96
Trace already off.

CPF3510
User exit program not found in specified library.

CPF3511
Trace already active.

CPF3512
Trace already off.

CPF3513
Cannot set Trace Off, trace started from another job.

CPF3521
Not enough storage for the trace table.

CPF3530
Conflicting entries in index QSERVICE.

CPF3542
Job not traced because it is being serviced.

CPF3548
Serviced job completed running.

CPF3675
Cannot allocate QSYS library.

CPF3909
Service command will not be processed.

CPF3918
Service request canceled.

CPF3925
Cannot open file &1.

CPF3936
Job being serviced ended before trace started.

CPF3950
Error message &2 received for file &1. Request ended.

CPF3951
File &1 cannot be overridden by file name &2.

CPF3957
Not authorized to use exit program library &2.

CPF3969
Error during close of file &1. Output may not be complete.

CPF6611
Error occurred during OUTFILE processing, trace ended.

CPF9810
Library &1 not found.

TRCREX (Trace REXX) Command Description

TRCREX Command syntax diagram

Purpose

The Trace REXX (TRCREX) command allows the user to turn the REXX interpreter tracing feature on or off from the command entry level or from the CL programming level.

Optional Parameter

SET Specifies the initial trace setting for the next REXX procedure that is run. This setting remains in effect unless changed through the REXX Trace instruction.

***RESULTS:** All clauses are traced before processing. The final results of an expression evaluation are traced. The values that were assigned during the PULL, ARG, and PARSE instructions are also shown. Tracing operates as if the Trace instruction is issued from within the REXX procedure.

***ALL:** All clauses are traced before processing. Tracing operates as if the Trace instruction is issued from within the REXX procedure.

***COMMANDS:** All host commands are traced before processing and any error return code is shown. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

***ERROR:** Host commands resulting in an error return code are traced after processing. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

***FAILURE:** Host commands resulting in a failure are traced after processing with the return code from the command. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

***INTERMEDIATES:** All clauses are traced before processing. Intermediate results during evaluation of expressions and substituted names are also traced. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

***LABELS:** Labels passed during processing are traced. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

***NORMAL:** Failing host commands are traced after processing. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

***OFF:** Nothing is traced. Tracing operates as if the Trace instruction is sent from within the REXX procedure.

Examples for TRCREX

Example 1: Tracing Host Commands

```
TRCREX   SET(*COMMANDS)
```

This command causes all commands in by the REXX procedure to be shown before they are to be run.

Example 2: Tracing Failing Host Commands

```
TRCREX   SET(*NORMAL)
```

This command causes all commands that result in a FAILURE condition to be shown. This command shows the normal setting for the REXX tracing operation.

Error messages for TRCREX

None

TRACEROUTE Command Description

TRACEROUTE Command syntax diagram

TRACEROUTE Command
For a description of the TRACEROUTE command, see the TRCTCPRTE Trace (TCP/IP Route) command description.

TRCTCPAPP (Trace TCP/IP Application) Command Description

TRCTCPAPP syntax diagram

Purpose

The Trace TCP/IP Application (TRCTCPAPP) command is used by service personnel when trace information needs to be collected for one of the following TCP/IP applications:

- FTP - Internal trace information can be collected for the FTP server. The information that can be collected for the FTP server may be filtered using remote network address or user profile.
- SMTP - Internal trace information can be collected for inbound or outbound connections. For inbound connections (*SMTPSVR) the trace information can also be filtered by remote network address or recipient mail address. For outbound connections (*SMTPCLT) the trace information can be filtered by recipient host name, recipient mail address or domain name service.
- TELNET and VTAPI - Telnet provides these qualifiers to help filter the collected information:
  - Remote network address
  - Local network address
  - Device description
  - Device types
  - Trace points
- Host Servers - Service personnel can collect trace information for several of the Client Access host servers: *CENTRAL, *DTAQ, *RMTCMD, *SIGNON, *NETPRT and *SVRMAP. » Trace information can also be collected for the database host server (*DATABASE). « The information that can be collected for these servers may be filtered using remote network address, local network address and trace points.
- DDM - Service personnel can collect trace information for Distributed Data Management (DDM) running over TCP/IP. The information may be filtered using remote network address, local network address and trace points.
- VPN - Service personnel can collect trace information for the Virtual Private Network (VPN) server. The information that can be collected may be filtered using the argument list and VPN server type.
- L2TP - Service personnel can collect trace information for Layer Two Tunneling Protocol (L2TP). The information that can be collected for these servers may be filtered using the remote network address.
- Certificate Services - Service personnel can collect trace information for the certificate services server. The information that can be collected for this server may be filtered using the certificate services type.
- PPP - Service personnel can collect trace information for a Point-to-point connection profile. The information that can be collected for this profile may be selected using the PPP connection profile name » and the argument list «. The user can specify what additional data (TCPTRCDTA) will be collected when ADLTRC(*TCPIP) is selected.
- QOS - Service personnel can collect trace information for the Quality of Service server. The information that can be collected for this server may be filtered using a trace type and an argument list.
- SNTP - Service personnel can collect trace information for the Simple Network Time Protocol client.
- HTTP - Service personnel can collect trace information for the HTTP server powered by Apache. The information that can be collected may be filtered by selecting an HTTP server instance and trace level.

- Directory Services - Service personnel can collect trace information for the directory services server. The information that can be collected may be filtered using an argument list.
- » Packet Rules - Service personnel can collect trace information for packet rules. The information that can be collected may be filtered using an argument list.«

Restrictions:

1. For a given application, only one trace instance can be active at a time.
2. This command is shipped with public *EXCLUDE authority.
3. To use this command, you must have either *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.
4. » The user must have *USE authority to the line, network interface, or network server to be traced.«

Required Parameter

APP Specifies the TCP/IP application.

***CENTRAL:** Specifies tracing for the central host server.

***CERTSRV:** Specifies tracing for certificate services. »

***DATABASE:** Specifies tracing for the database host server. «

***DDM:** Specifies tracing for the Distributed Data Management (DDM) server.

***DIRSRV:** Specifies tracing for directory services.

***DTAQ:** Specifies tracing for the data queue host server.

***FTP:** Specifies tracing for the File Transfer Protocol (FTP) server(s).

***HTTP:** Specifies tracing for the HTTP server powered by Apache.

***L2TP:** Specifies tracing for Layer Two Tunneling Protocol (L2TP).

***NETPRT:** Specifies tracing for the network print host server.

***NTP:** Specifies tracing for the Simple Network Time Protocol (SNTP) client. »

***PKTRULES:** Specifies tracing for packet rules. «

***PPP:** Specifies tracing for the Point-to-point protocol (PPP).

***QOS:** Specifies tracing for the Quality of Service (QOS) server.

***RMTCMD:** Specifies tracing for the remote command host server.

***SIGNON:** Specifies tracing for the signon host server.

***SMTPCLT:** Specifies tracing for the SMTP client job(s) handling outbound mail processing connections.

***SMTPSVR:** Specifies tracing for the Simple Mail Transfer Protocol (SMTP) server job(s) handling inbound mail processing connections.

***SVRMAP:** Specifies tracing for the port mapper host server.

***TELNET:** Specifies tracing for the TELNET server.

***VPN:** Specifies tracing for the Virtual Private Network (VPN) server.

***VTAPI:** Specifies tracing for the virtual terminal application programming interfaces.

Optional Parameters

SET Specifies whether the collection of trace information starts, stops, or status is presented.

***ON:** The collection of trace information is started.

***OFF:** The collection of trace information is stopped, and the trace information is written to spooled printer files of the user. For PPP traces, the trace files are also included in the OUTQ for the designated PPP profile.

***END:** Tracing is ended and all trace information is deleted. No trace information output is created.

***CHK:** The status of tracing for the specified application is checked. Messages are returned indicating whether or not tracing is active for the specified TCP/IP application, the command parameters specified from the last time that TRCTCPAPP was started for this application, and other information related to the collection of trace information.

MAXSTG

Specifies the maximum amount of storage in kilobytes (K) used for collected trace information. Values can range from 1-16,000.

***APP:** Each application type defines a default buffer size.

*FTP - 4096K bytes per job

*SMTPCLT or *SMTPSVR - 4096K bytes per job

*TELNET or *VTAPI - 16,000K bytes per job

Host Servers - 16,000K bytes per job

*DDM - 16,000K bytes per job

*VPN - 16,000K bytes per job

»

*PKTRULES - 16,000K bytes per job«

*L2TP - 4096K bytes per job

*CERTSRV - 16,000K bytes per job

*PPP - 4096K bytes per job.

*QOS - 4096K bytes per job

*NTP - 4096K bytes per job

*HTTP - 16,000K bytes per job

*DIRSRV - 300K bytes per job

maximum-K-bytes: Specify the maximum amount of storage, in K-bytes, used to store trace records (one K equals 1024 bytes).

TRCFULL

Specifies whether the trace records wrap (replace oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

***WRAP:** When the trace buffer is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing stops when the trace buffer is full of trace records.

ADLTRC

Specifies additional trace(s) to be started. When the TRCTCPAPP command is invoked interactively, the user will be prompted for any options to change on each of the selected traces. » This parameter is valid for all applications. «

***NONE:** No additional trace will be included.

***CMNTRC:** A communications trace will be included in the trace information for the specified application.

Note: Due to resource limitations of the I/O hardware, multi-connection PPP profiles may not produce trace data for every connection started by the PPP profile.

***TCPIP:** A single TCP/IP component trace will be included in the trace information for the specified application.

***SRCSINK:** A source/sink component trace will be included in the trace information for the specified application.

» **TRCPGM**

Specifies the name of a program to call for user defined trace commands and procedures.

For SET(*ON), the trace program will be called:

- Before the application trace starts.
- After the communications and Licensed Internal Code (LIC) traces, if requested, start.

For SET(*OFF), the trace program will be called:

- Before the LIC traces, if requested, end.
- After the communications trace, if requested, ends.
- After the application trace ends.

For SET(*END), the trace program will be called:

- After the LIC and communications traces, if requested, end.
- After the application trace ends.

This parameter is valid for all applications.

When the TRCTCPAPP command detects an error from the trace program, it will send a TCP4537 diagnostic message.

There are two input parameters and one output parameter associated with the trace program. The three parameters are required:

- 1 Trace option setting Input Char(10)
- 2 Application Input Char(10)
- 3 Error detected Output Char(10)

The possible values for the “Trace option setting” parameter are:

- *ON, The collection of trace information is started.
- *OFF, The collection of trace information is stopped and the trace information is written to spooled printer files of the user.
- *END, Tracing is ended and all trace information is deleted. No trace information output is created.

The possible values for the “Application” parameter are the same as the values for the APP parameter on the TRCTCPAPP command.

The possible value for the “Error detected” parameter is:

- *ERROR, Error detected by customer trace program.

***NONE:** No user supplied trace program will be called.

The possible library values are:

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

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

trace-program-library: Specify the name of the library where the program is located.

trace-program: Specify the name of the user supplied program to call.«

TITLE Specifies the title that is printed on each page of the spooled file which contains the collected trace information. This parameter is only valid when SET(*OFF) is specified. This parameter can be up to » 50 « characters.

***DFT:** The default trace description title “TRCTCPAPP Output” is used.

’trace-title’: If entered, this string is placed on each page of the trace output spooled file.

USER Only trace information associated with a specific job profile will be collected. This parameter is only valid when APP(*FTP) is specified.

user-profile-name: Specify the user profile for which trace information is to be collected.

MAILADR

Only trace information associated with a specific recipient mail address will be collected. This parameter is only valid when APP(*SMTPSVR) or APP(*SMTPCLT) is specified.

recipient-mail-address: The recipient address (up to 255 characters) must have the following format:

“userid@abc.def.com”

HOST Specifies the recipient host name for which trace information is collected. This parameter is only valid when APP(*SMTPCLT) is specified.

recipient-host-name: The recipient host name (up to 255 characters) must have the following form:

“abc.def.com”

RMTNETADR

The user may limit the amount of information collected by entering an address family, remote TCP/IP address, subnet mask, and port number. This parameter is only valid when APP(*FTP), APP(*SMTPSVR), APP(*CENTRAL), APP(*DTAQ), APP(*RMTCMD), APP(*SIGNON), APP(*NETPRT), APP(*SVRMAP), APP(*DDM), APP(*TELNET), APP(*VTAPI), APP(*L2TP) or » APP(*DATABASE) « is specified. Note that the only valid filter for L2TP is the IP Address element.

Element 1: Address Family

***INET:** Filter for AF_INET address family.

Element 2: IP Address

IP-address: Specify the remote TCP/IP address for which trace information is to be collected.

Element 3: Subnet Mask

255.255.255.255: Tracing will be done for only the IP address specified as the second element of this parameter.

subnet-mask: Specify the subnet mask for which trace information is to be collected.

Element 4: Port Number

***ANY:** The TCP/IP port number defaults to *ANY which implies traffic associated with any port on the remote system (and qualified by the IP address and subnet mask) will be traced.

TCP/IP-port-number: If the user wants to specify the port, the subnet mask must also be specified.

LCLNETADR

The user may limit the amount of information collected by entering an address family, local TCP/IP address, subnet mask, and port number. This parameter is only valid when APP(*TELNET), APP(*VTAPI), APP(*CENTRAL), APP(*DTAQ), APP(*RMTCMD), APP(*SIGNON), APP(*NETPRT), APP(*SVRMAP), APP(*DDM) or » APP(*DATABASE) « is specified.

Element 1: Address Family

***INET:** Filter for AF_INET address family.

***UNIX:** Filter for AF_UNIX address family.

Element 2: IP Address or UNIX path

IP-address: When *INET specified for element 1 of this parameter, specify the local TCP/IP address for which trace information is to be collected.

UNIX-path: When *UNIX specified for element 1 of this parameter, specify the UNIX path for which trace data is to be collected.

Element 3: Subnet Mask

255.255.255.255: Tracing will be done for only the IP address specified as the second element of this parameter.

subnet-mask: Specify the subnet mask for which trace information is to be collected.

Element 4: Port Number

***ANY:** The TCP/IP port number defaults to *ANY which implies traffic associated with any port on the local system (and qualified by the IP address and subnet mask) will be traced.

TCP/IP-port-number: If the user wants to specify the port, the subnet mask must also be specified.

DEVD The user may limit the amount of information collected by entering a device description name. Once the device description is associated with a given TELNET or VTAPI session, all trace information associated with it will be collected. This parameter is only valid when APP(*TELNET) or APP(*VTAPI) is specified.

device-description-name: Specify the name of a device description for which trace information is to be collected.

generic-device-description-name:* Specify the generic name for device descriptions for which trace information is to be collected. A generic name is a character string of one or more characters followed by an asterisk (*); for example, CMN*. If a generic name is specified, then all device descriptions with names that begin with the generic name, and for which the user has authority, will have trace information collected.

DEVTYPE

One or more valid device types may be specified. Only the trace information associated with activity for those devices will be traced. If *DSP or *PRT is specified, no other values may be entered for this parameter. This parameter is only valid when APP(*TELNET) or APP(*VTAPI) is specified.

***DSP:** The information collected is only for display device types.

***PRT:** The information collected is only for printer device types.

device-type: The information collected is only for the specified device types. Up to six types may be specified. The valid types include: 5251, 5291, 5292, 3196, 3488, 3487, 3179, 3180, 5555, 3477, 3277, 3278, 3279, V100, 3812 and 5553.

TRCPNT

You can limit the trace points that are placed in the trace buffer by entering the list of those trace points for this parameter. Up to 12 trace points may be specified. This parameter is only valid when APP(*TELNET), APP(*VTAPI), APP(*CENTRAL), APP(*DTAQ), APP(*RMTCMD), APP(*SIGNON), APP(*NETPRT), APP(*SVRMAP), APP(*DDM) or » APP(*DATABASE) « is specified.

trace-point-identifier: Each trace point may be up to 8 characters. For TELNET/VTAPI trace points, specify ’TG#xxxxx’, ’TG+xxxxx’ or ’TG-xxxxx’ where ’xxxxx’ defines the specific trace point. The following TELNET/VTAPI trace points can also be specified: TGTELM, TGTELO, TGEXCP, TGREQPO, TGRIO, TGRPO, TGUTIL, TGVTERM, TGVTIN, TGVTINI, TGVTM or TGVTOUT. For host/DDM server trace points, specify ’Qcccxxxx’ where ’ccc’ is the component ID of the host/DDM server and ’xxxx’ defines the specific trace point.

ARGLIST

Only trace information associated with this specific argument list (up to 255 characters) will be included in the trace information collected. The argument list contains data like the debug level and special trace requests. This parameter is only valid when APP(*VPN), APP(*QOS), » APP(*PKTRULES) « , APP(*PPP) or APP(*DIRSRV) is specified.

argument-list: Specify the argument list (up to 255 characters).

QoS allows the following argument list values:

lvl=1: The lvl=1 argument logs errors that are associated with system operations. One example might be that the system is out of memory. The result of these types of errors is that the QoS server will not be able to run.

lvl=2: The lvl=2 argument includes all lvl=1 information. In addition, the lvl=2 argument logs internal errors identified with the operation of the QoS server. The usual cause of these types of errors is that unexpected errors have been encountered in a server operation. A lvl=2 error is considered a condition for an APAR.

lvl=3: The lvl=3 argument includes all lvl=1 and lvl=2 information. In addition, the lvl=3 argument logs the basic operational activities of the QoS server. Examples might be the loading of rules or the sending of a STRTCPSVR SERVER(*QOS) RESTART(*QOS) command.

lvl=4: The lvl=4 argument includes all lvl=1, lvl=2 and lvl=3 information. In addition, the lvl=4 argument logs all traced activities of the QoS server.

VPNSVR

Specifies whether the trace information is to be collected for only the VPN key manager or the VPN connection manager. If no value is specified for this parameter, trace information is to be collected for both the VPN key manager and the VPN connection manager. This parameter is only valid when APP(*VPN) is specified.

***KEYMGR:** Filtering of trace information is done to include the VPN key manager.

***CNNMGR:** Filtering of trace information is done to include the VPN connection manager.

CERTTYPE

Only trace information associated with a specific certificate services type will be included in the trace information collected. This parameter is only valid when APP(*CERTSRV) is specified. »

***ALL:** No filtering of trace information is done for certificate services type. «

***DCM:** Filtering of trace information is done to include only the DCM certificate services type.

***KEYMGR:** Filtering of trace information is done to include only the VPN key manager certificate services type.

***SSL:** Filtering of trace information is done to include only the SSL certificate services type.

***OBJSIGN:** Filtering of trace information is done to include only the OBJSIGN certificate services type.

***OTHER:** Filtering of trace information is done to include only a certificate services type not listed above.

DNS Specifies whether only trace information associated with domain name service (DNS) resolution will be collected. This parameter is only valid when APP(*SMTPCLT) is specified.

***NO:** No filtering of trace information is done for DNS resolution.

***YES:** Trace information includes only trace points associated with DNS resolution.

PPPCNNPRF

Trace information associated with a specific PPP connection profile will be collected. The default trace information provided is one joblog and one connection dialog spooled file (containing the PPP dialog trace) for each connection started by the PPP connection profile, one copy of the PPP profile settings, and one copy of the line description used by the profile. When selected by the user there could also be one SRCSINK component trace per connection, one Communications trace per connection, and a single TCPIP component trace. This parameter is required when APP(*PPP) is specified.

connection-profile-name: Specify the PPP connection profile for which trace information is to be collected.

TCPTRCDTA

Specifies what additional data will be collected when ADLTRC(*TCPIP) is selected. This parameter is only valid when APP(*PPP) is specified. If APP(*PPP) is specified and ADLTRC(*TCPIP) is not specified, this parameter will be ignored.

***PPPALL:** All data on the PPP connection will be traced.

***LCPNCP:** Only LCP and NCP data on the PPP connection will be traced.

QOSTRCTYPE

Only trace information associated with a specific QOS trace type will be included in the trace information collected. This parameter is only valid when APP(*QOS) is specified.

***ALL:** Filtering of trace information is done to include both servers.

***POLICYD:** Filtering of trace information is done to include the QOS policy server.

***RSVPD:** Filtering of trace information is done to include the RSVP (Resource reSerVation Protocol) server.

HTTPSVR

This parameter will determine which HTTP server instance to trace. It is only valid and required when APP(*HTTP) is specified.

TRCLVL

Specifies the level of granularity of the service trace. This parameter is only valid when APP(*HTTP) is specified.

***ERROR:** The service trace will contain trace records for all error return codes or exception conditions.

***INFO:** The service trace will contain *ERROR level trace records as well as trace records for entry and exit points from application level API's and API parameters.

***VERBOSE:** The service trace will contain *INFO level trace records as well as trace records for debugging control flow or data corruption.

» **PKTTRCPNT**

Specifies a keyword that represents trace point values which will be displayed when the Trace Internal (TRCINT) panel is displayed. This parameter is only valid when APP(*PKTRULES) and ADLTRC(*TCPIP) are specified.

***TRAFFIC:** The following trace points for packet filter evaluation will be displayed on the Trace Internal panel: 8110-8111, 8120-8123 and 8420.

***LOAD:** The following trace points for the audit and load of rules will be displayed on the Trace Internal panel: 8100-8105 and 8430-8438.

CFGOBJ

Specifies the name of the configuration object to trace. The object can be either a line description, a network interface description, or a network server description. This parameter is only valid when ADLTRC(*CMNTRC) is specified.

CFGTYPE

Specifies the type of configuration description to trace. This parameter is only valid when ADLTRC(*CMNTRC) is specified.

***LIN:** The type of configuration object is a line description.

***NWI:** The type of configuration object is a network interface description.

***NWS:** The type of configuration object is a network server description.«

Examples for TRCTCPAPP »

Example 1: Starting Trace for Database

```
TRCTCPAPP APP(*DATABASE) SET(*ON) TRCPNT(QZDA1050 QZDA1060)
          LCLNETADR(*INET '9.130.69.22' '255.255.255.255' 8471)
          ADLTRC(*CMNTRC) TRCPGM(PGMLIB/PROG1) CFGOBJ(TESTLIN)
          CFGTYPE(*LIN)
```

This will start tracing for the database host server. Tracing information associated with the AF_INET address family, a local TCP/IP address of 9.130.69.22, a subnet mask of 255.255.255.255, port number of 8471 and trace points of QZDA1050 and QZDA1060 will be collected. A communication trace will be included in the trace information. TESTLIN is the name of the configuration object to trace. This object is a line (*LIN) description. Trace program PROG1 in library PGMLIB, with its user-defined trace commands and procedures, will be called. Tracing for the other TCP applications is not affected.

Example 2: Check Status of Database Tracing

```
TRCTCPAPP APP(*DATABASE) SET(*CHK)
```

This command is used to check the status of the tracing for the database host server job. Assume that the last command entered was from “Example 1”. The format of the response to this command would be a set of messages that would look similar to the following:

```
TCP45B7 TRCTCPAPP APP(*DATABASE) SET(*ON) TRCPNT(QZDA1050 QZDA1060)
                  LCLNETADR(*INET '9.130.69.22' '255.255.255.255' 8471)
                  MAXSTG(*DFT) TRCFULL(*WRAP) ADLTRC(*CMNTRC)
                  TRCPGM(PGMLIB/PROG1) CFGOBJ(TESTLIN) CFGTYPE(*LIN)
TCP45B1 Tracing active for *DATABASE at 20:15:14 on 03/15/01 by
        043432/TRCUSER/QPADEV000B.
TCP45B2 Data capture begun for *DATABASE.
TCP45B3 Data buffer wrapped for *DATABASE.
```

Example 3: Ending Database Connection Tracing

```
TRCTCPAPP APP(*DATABASE) SET(*OFF)
```

This command first ends any currently active application trace for the database host server, followed by ending the TCP/IP component trace. If tracing was active, output trace records are formatted and placed into a spool file. A similar message will be found in the user’s joblog:

```
TCP45B8 Trace data for application DATABASE formatted: QZDA001915.
```

If tracing is not active, then the following message will be returned to the user:

```
TCP4580 Tracing off, SET(*OFF) not valid.
```

Example 4: Starting Trace for Packet Rules

```
TRCTCPAPP APP(*PKTRULES) SET(*ON) ARGLIST('DebugLvl=1 TraceLvl=2')
          ADLTRC(*TCPIP) PKTTRCPNT(*LOAD)
```

This will start tracing for packet rules. Tracing information associated with the specific argument list will be collected. A component trace will be included in the trace information, using trace points of 8100-8105 and 8430-8438. Tracing for the other TCP applications is not affected. «

Example 5: Starting Trace for FTP

```
TRCTCPAPP APP(*FTP) SET(*ON)
          RMTNETADR(*INET '9.130.69.16' '255.255.255.255' 5)
```

This will start tracing for the FTP server. Tracing information associated with the AF_INET address family, a remote TCP/IP address of 9.130.69.16, a subnet mask of 255.255.255.255 and port number of 5 will be collected. Tracing for the other TCP applications is not affected.

Example 6: Starting Trace for TELNET

```
TRCTCPAPP APP(*TELNET) SET(*ON) DEVD(QPADEV*)
```

This will start tracing for the TELNET server. Trace information will be collected for all device descriptions with names that begin with “QPADEV”. The user must have authority to these specific device descriptions. Tracing for the other TCP applications is not affected. »

Error messages for TRCTCPAPP

ESCAPE Messages

TCP4595
Trace not started.

«

TRCTCPRTE (Trace TCP/IP Route) Command Description

TRCTCPRTE Command syntax diagram

Purpose

The Trace TCP/IP Route (TRCTCPRTE) command, also known as TRACEROUTE, traces the route of IP packets to a user-specified destination system. The route can involve many different systems along the way. Each system along the route is referred to as a **hop**. You can trace all hops along the route or specify the starting and ending hops to be traced.

The route is traced by sending packets (called **probes**) to the destination system. Each probe contains an upper limit on the number of systems the probe can pass through.

» Note: In IP Version 6, ’Time To Live’ (TTL) is called the ’hop limit’. «

A route is traced by successively incrementing the TTL (or hop limit) of the probe packets by one. The trace ends when either a probe response is received from the destination system or when the probe Time To Live value equals the maximum allowed.

Responses from the probe packets are sent as messages to the job log or as queue entries to a user-specified data queue.

Required Parameter

RMTSYS
Specifies the remote system name (255 characters) or IP address of the destination system. » Either a valid IP Version 4 or IP Version 6 address will be accepted. «

remote-system: Specify the remote system name or address.

Optional Parameters

RANGE
Specifies the range of hop systems from which probe responses are expected. Each probe specifies a Time To Live (or hop limit) integer value. This Time To Live (TTL) value is the maximum number of hops the probe can traverse. For example, a probe packet with a TTL of 3 can pass through at most 3 hop systems before the hop system discards the probe and sends information back to the system from which the probe originated.

Element 1 specifies the first TTL value sent in probe packets. Element 2 specifies the last TTL value sent in probe packets. Trace information is generated from each hop system which discards a probe packet because the TTL value in the probe is reached or when the destination system is reached.

Element 1: Starting Probe TTL » (hop limit) «

1: The default starting hop is 1.

starting-hop: Specify the first hop limit TTL number used for probe packets. Valid values range from 1 to 255.

Element 2: Ending Probe TTL » (hop limit) «

30: The default ending hop is 30.

ending-hop: Specify the maximum number of hops a probe can traverse to reach the destination system. Valid values range from 1 to 255.

PROBES
Specifies the number of probe packets sent to each hop system for each probe TTL » (hop limit) « value in the range specified by the RANGE parameter.

3: The default number of probes is three.

number-of-probes: Specify the number of probes to send. Valid values range from 1 to 64.

WAITTIME
Specifies the maximum time, in seconds, to wait for a response from a hop system to each probe.

3: The default is to wait up to 3 seconds for a response.

response-waittime: Specify the maximum time in seconds to wait for a response. Valid values range from 1 to 120.

PKTLEN
Specifies the total length, in bytes, of the IP packet sent for each probe.

40: The default probe packet length.

packet-length: Specify the probe IP packet length, in bytes. Valid values range from 40 to 65535.

OUTPUT
Specifies where the results obtained from sending the probe packets is sent. Information is sent for each hop until the destination system is reached, including hop count, average round-trip time, IP address of the hop and host name of the hop.

***MSG:** Results are output as messages sent to the job log of the job in which the command is issued.

***VERBOSE:** Results are output as messages sent to the job log of the job in which the command is issued. All responses received are displayed. Results are not limited to ICMP TIME_EXCEEDED and PORT_UNREACHABLE responses.

***DTAQ:** Results from probes are placed on the data queue specified by the DTAQ parameter.

DTAQ Specifies the name and library of the data queue on which entries are placed. When a data queue is specified, messages are not sent to the job log unless an error occurs.

Each queue entry contains the response to a probe if one was received or indicates that no probe response was received. The specified data queue must have a queue entry length of at least 32 characters and must exist when the command is issued.

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

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.

» **ADRVERFMT**

Specifies which IP address version protocol (IP Version 4 or IP Version 6) is used to resolve host names specified by RMTSYS.

***CALC:** The address family will be 'calculated' (determined) based on the address entered by the RMTSYS parameter. TRCTCPRTE (TRACEROUTE) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution will then be used in order to resolve the IP address.

***IP4:** Explicitly use the IP Version 4 protocol addressing method.

***IP6:** Explicitly use the IP Version 6 protocol addressing method. «

LCLINTNETA

Specifies how the source IP address in the probe packet is chosen.

***ANY:** The source IP address in the probe packets is chosen by the system. The system may use any active local interface which can reach the remote system.

source-IP-address: The local interface to use as the source IP address.

RMTPORT

Specifies the base UDP port number used in probes.

33434: The default base UDP port number.

base-remote-port: Specify the base UDP port number used in probes. Valid values range from 1 to 65535.

NAMELOOKUP

Specifies whether IP addresses will be resolved to the host name.

***YES:** The address will be resolved to the host name.

***NO:** The address will not be resolved to the host name.

PROBEPCL

Specifies the protocol used when sending probe packets.

***ICMP:** The probes sent to the destination system are ICMP (Internet Control Message Protocol) Echo Request packets.

***UDP:**The probes sent to the destination system are UDP (User Datagram Protocol) packets.

FRAGMENT

Specifies how the setting of the “Do Not Fragment” option in the IP header of the probe packet is determined.

***TCPA:** The iSeries system sets the option based on the setting of the IP Path MTU Discovery TCP/IP attribute.

Note: Use the Change TCP/IP Attributes (CHGTCPA) command to change the value of this attribute.

***NO:** The “Do Not Fragment” option is always not set in probe packets.

***YES:** The “Do Not Fragment” option is always set in probe packets.

Examples for TRCTCPRTE

Example 1: Trace Entire Route

```
TRCTCPRTE RMTSYS('130.14.3.5')
```

This command traces the entire route between the local iSeries system and the destination system whose IP address is ’130.14.3.5’. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP Echo Request packet. Results received are sent as messages to the job log.

Example 2: Trace Partial Route

```
TRCTCPRTE RMTSYS('AAA.BBBB.COM')
  RANGE(3 7) PROBES(5) PROBEPCL(*UDP)
  OUTPUT(*DTAQ) DTAQ(MYLIB/MYDATAQ)
```

This command traces the route between the local iSeries system and the destination system whose host name is ’AAA.BBBB.COM’. Five probe packets will be sent for the starting range value of 3. Each probe will be a UDP packet inside an IP packet that is 40 bytes long. Each of these 5 probes will specify a TTL of 3. If system AAA.BBB.COM can be reached by passing through at most 2 hop systems then the trace will terminate at this point.

If system AAA.BBB.COM is futher than 2 hops, another set of 5 probe packets will be sent to the destination AAA.BBB.COM. Each of these 5 probes will specify a TTL of 4. This is repeated until either system AAA.BBB.COM responds to a probe or 5 probes with a TTL of 7, the ending range value, are sent. Any results received are placed as queue entries on data queue MYDATAQ in library MYLIB. »

Example 3: Trace Route with an IP Version 6 address

```
TRCTCPRTE RMTSYS('1:2:3:4:5:6:7:8')
```

This command traces the entire route between the local iSeries system and the destination system whose IP address is ’1:2:3:4:5:6:7:8’. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

Note: A colon (’:’) found in RMTSYS signifies an IP Version 6 address and will cause an ICMP6 echo request packet to be generated.

Example 4: Trace Route with an IP Version 6 host name

```
TRCTCPRTE RMTSYS('IP6HOST')
```

This command traces the entire route between the local iSeries system and the destination system whose host name is ’IP6HOST’. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

The default “Address version format” is *CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to trace the route.

Example 5: Trace Route with an IP Version 6 host name and explicitly use IP Version 6 host name resolution

```
TRCTCPRTE RMTSYS('IP6HOST') ADRVERFMT(*IP6)
```

This command traces the entire route between the local iSeries system and the destination system whose host name is ’IP6HOST’. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

This example differs from example 4 in that only a valid IP version 6 resolved address, for IP6HOST, will be used when attempting to trace the route. «

Error messages for TRCTCPRTE

***ESCAPE Messages**

TCP3250
DTAQ parameter value required with OUTPUT(*DTAQ).

TCP3251
DTAQ parameter not valid when OUTPUT(*MSG) specified.

TCP3252
Starting range value greater than range limit.

TFRBCHJOB (Transfer Batch Job) Command Description

TFRBCHJOB Command syntax diagram

Purpose

The Transfer Batch Job (TFRBCHJOB) command transfers a batch job to the specified job queue. The job queue does not have to be allocated to an active subsystem at the time of the batch job transfer. The batch job that is transferred is the one in which this TFRBCHJOB command is issued. Routing data and request data can be specified for the batch job when it is transferred. The routing data specified is processed in the subsystem in which the job queue is active. The request data is placed after all other request data for the job. The transferred batch job resumes running the request data immediately following the TFRBCHJOB command.

If the TFRBCHJOB command is issued in a CL program, all subsequent commands in the CL program are bypassed. If objects that are allocated to the current routing step or files that are open in the current routing step are needed in the new routing step, they must be allocated or opened again after the new routing step has been started.

Note: If you are working in a System/36 environment, the TFRBCHJOB command does not transfer the System/36 environment to the new job.

A batch job transferred to a job queue by the TFRBCHJOB command exists through an initial program load (IPL) if the batch job was residing on the job queue at the time the system was powered down; the job's temporary objects exist through a power down. A batch job's temporary objects are destroyed during the power down if the job was transferred by the Transfer Job (TFRJOB) command.

The QTEMP library of a batch job that has been transferred by the TFRBCHJOB command is always empty when the next routing step is started. Caution must be used with the library list in conjunction with a batch job that was transferred to a job queue by the TFRBCHJOB command. The TFRBCHJOB function saves the library list to recover the job on a job queue if an IPL occurs. When the routing step for the transferred batch job is started, the libraries in the saved library list must exist in the system or the job's routing step ends.

Restrictions: (1) The user must have add and read authority for the job queue and operational authority for the subsystem to which the job queue is allocated. (2) The job being transferred must be a batch job that started from a job queue. (3) The TFRBCHJOB command is not allowed to run in a batch communications job (a batch job that was started as the result of a program start request) or a batch immediate job. Also, interactive, auto-start, reader, writer, and system jobs cannot be transferred by the TFRBCHJOB command.

Required Parameter

JOBQ Specifies the qualified name of the job queue to which the batch job is transferred.

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

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

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

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

job-queue-name: Specify the name of the job queue to which the batch job is transferred.

Optional Parameters

RTGDTA

Specifies the routing data used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program where the job runs.

QCMDB: This routing data matches a routing entry in the IBM-supplied subsystem description, which starts a routing step processed by the IBM-supplied control language processor QCMD.

***RQSDTA:** Up to 80 characters of the request data specified in the RQSDTA parameter of this command are used as the routing data for the routing step. If this value is used, RQSDTA(*RTGDTA or *NONE) cannot be specified.

’routing-data’: Specify the character string that is used as the routing data for starting the routing step. Up to 80 characters can be specified, enclosed in apostrophes if blanks or special characters are included.

RQSDTA

Specifies the request data that is added to the end of the job’s message queue for use by the new routing step.

***NONE:** No request data is placed in the job’s message queue.

***RTGDTA:** The routing data specified in the RTGDTA parameter is placed at the end of the job’s message queue. If this value is used, RTGDTA(*RQSDTA) cannot be specified.

’request-data’: Specify the character string that is placed at the end of the job’s message queue for use by the new routing step or some subsequent routing step in the job. Up to 256 characters can be specified, enclosed in apostrophes if blanks or special characters are included. When a CL command is specified, it must be enclosed in single apostrophes, and where apostrophes would normally be used in the command, double apostrophes must be used.

Example for TFRBCHJOB

```
TFRBCHJOB  JOBQ(QGPL/QCTL)  RTGDTA(APPLICS)
```

This command transfers the batch job where the command is entered to the QCTL job queue that is in the QGPL library. The job is routed using the routing data APPLICS. The job must be a batch job.

Error messages for TFRBCHJOB

***ESCAPE Messages**

CPF1288
Job queue &1 in library &2 damaged.

CPF1289
Transfer job is not allowed.

CPF1291
Job &3/&2/&1 cannot be transferred.

CPF1368
&1 not authorized to job queue &2 in library &3.

CPF1369
Job queue &1 in &2 not found.

CPF1370
Job queue &1 in &2 not accessible.

CPF1372
Job not transferred. Job currently being ended.

CPF1376
Library on library search list deleted.

CPF1377
Library on library search list damaged.

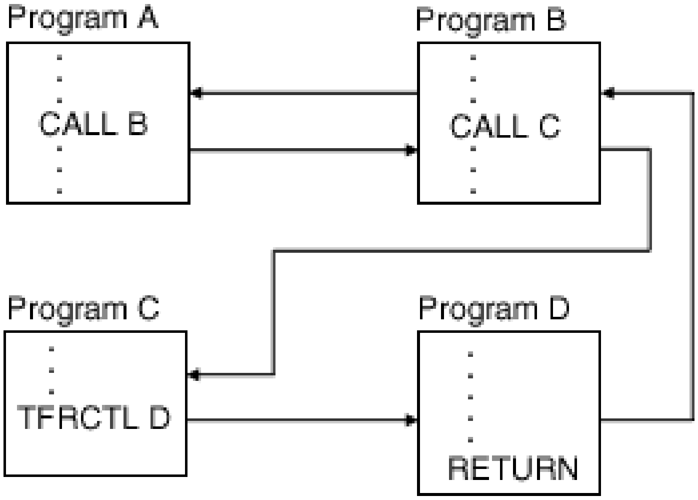
TFRCTL (Transfer Control) Command Description

TFRCTL Command syntax diagram

Purpose

The Transfer Control (TFRCTL) command calls the program specified on the command, passes control to it, and removes the transferring program from the return stack. Because the transferring program is removed from the call stack, control does not return to it when the called program returns control. Instead, control is returned to the command following the last call to the transferring program.

The following diagram shows the operation of both the CALL and TFRCTL commands. First, Program A calls Program B. Program B then calls Program C, which in turn transfers control (using the TFRCTL command) to Program D. In transferring control to Program D, Program C is removed from the call stack; therefore, Program D returns to Program B (to the command following the most recent call to the transferring program, Program C). Finally, Program B returns to Program A.



If the transferring program was created with USRPRF(*OWNER), the authority does not transfer. However, if a program created with USRPRF(*OWNER) calls a program that transfers control, the authority *does* transfer because the program that called the transferring program remains in the call stack.

Optionally, the transferring program passes parameters to the program being called. The storage space used by the CL variables in the transferring program is freed and is available for use by the program receiving control.

The parameter values must be passed in CL variables. Values cannot be passed as constants, as null parameters (that is, parameters whose values are null, specified by *N), as lists of values, or as CL variables that were not specified as parameters on the PGM command that identified the start of the transferring program.

The transferring program can only pass CL variables that were previously passed to it. No CL variable can be passed that exists within the transferring program itself. Up to 40 parameters can be passed to the called program. The parameters passed must agree in type, length, number, and order with those expected by the receiving program, as specified in the PARM parameter of the PGM command.

Shared files remain open if control is transferred to a program from a high-level language program or from the CL program that opened the files.

Restrictions:

1. This command is valid only within CL programs.
2. The user must have use authority or one of the data authorities for the program to which control is being transferred.

Required Parameter

PGM Specifies the qualified name of the program that receives control from the program transferring control.

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

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

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

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

program-name: Specify the name of the program that receives control.

Optional Parameter

PARM Specifies the names of one or more CL variables that are passed to the program that is receiving control. The variables passed can only be parameters that were passed to the program currently transferring control. Up to 40 variables can be specified. Specify a CL variable name for each of the values being passed by the program transferring control. The parameter values are received in the receiving program in the order in which they were specified on the TFRCTL command that calls the program.

The value *N cannot be specified as a value being passed because a null value cannot be passed to another program.

Example for TFRCTL

```
TFRCTL   PGM(PROGA)  PARM(&PARM1)
```

This command transfers control to the program PROGA and passes the parameter &PARM1 to it. The parameter &PARM1 must previously have been passed to the program issuing this command.

Error messages for TFRCTL

***ESCAPE Messages**

CPF0805
Error found when program &1 in &2 started.

CPF0809
Transfer control (TFRCTL) to C program not allowed.

TFRJOB (Transfer Job) Command Description

TFRJOB Command syntax diagram

Purpose

The Transfer Job (TFRJOB) command transfers a job to the specified job queue that is run in the subsystem in which that queue is active. The job that is transferred is the one in which this TFRJOB command is issued. The specified job queue is normally in a different subsystem than the one in which the job is currently. If the job being transferred is an interactive job, it is given the highest priority on the job queue. New routing data and request data can be specified for the job when it is transferred. If objects that were allocated to the previous routing step or files that were open in the previous routing step are needed in the new routing step, they must be allocated or opened again.

Note: Running this command in a batch job causes loss of spooled inline files because they cannot be accessed in the new routing step.

If the target subsystem is ended (by running the End Subsystem (ENDSBS) command, the End System (ENDSYS) command, or the Power Down System (PWRDWNSYS) command) while an interactive transferring job is on a job queue, the job is canceled as part of subsystem ending.

Because a Power Down System (PWRDWNSYS) command inhibits new jobs or routing steps from being started by any subsystem, a batch job transferred to a job queue (by the TFRJOB command) is not completed before the system is powered down. The temporary objects associated with a transferring job (such as the library list, the QTEMP library, and all objects in it) are destroyed during the PWRDWNSYS command, so that during a re-initial program load (IPL), the system is unable to restore the job to its previous state. During re-IPL, the system removes the job from the job queue and produces its job log.

Restrictions: The user must have read and add authority for the job queue and for the subsystem to which the job queue is allocated. If the job being transferred is an interactive job, the following restrictions apply:

- The job queue on which the job is placed must be associated with an active subsystem.
- The work station associated with the job must have a corresponding work station entry in the subsystem description associated with the new subsystem.
- The work station associated with the job must not have another job associated with it that has been suspended by means of the Sys Req (system request) key. The suspended job must be canceled before the Transfer Job command can be run.
- The job must not be a group job.
- The job must not be a communications batch job (started as a result of a program start request), unless it meets one of the following criteria:
  - It was started from an APPC communications device
  - The session on the communications device has ended

Required Parameter

JOBQ Specifies the qualified name of the job queue to which the job is transferred.

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

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

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

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

job-queue-name: Specify the name of the job queue to which the job is transferred.

Optional Parameters

RTGDTA

Specifies the routing data used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program in which the job runs.

QCMDI: This routing data matches a routing entry in the IBM-supplied QINTER subsystem description, which starts a routing step processed by the IBM-supplied control language processor, QCMD, in the QSYS library.

***RQSDTA:** Up to 80 characters of the request data, specified in the RQSDTA parameter of this command, are also used as the routing data for the routing step.

’routing-data’: Specify the character string that is used as the routing data for starting the routing step. Up of 80 characters can be entered, enclosed in apostrophes, if necessary.

RQSDTA

Specifies the request data that is added to the end of the job’s message queue for use by the new routing step.

***NONE:** No request data is placed in the job’s message queue.

***RTGDTA:** The routing data specified in the RTGDTA parameter is also placed at the end of the job’s message queue.

’request-data’: Specify the character string that is placed at the end of the job’s message queue for use by the new routing step or some subsequent routing step in the job. Up to 256 characters can be entered, enclosed in apostrophes, if necessary. When a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used *inside* the command, double apostrophes must be used instead.

Example for TFRJOB

```
TFRJOB  JOBQ(QSYS/QCTL)  RTGDTA(APPLICS)
```

This command transfers the job in which the command is entered to the QSYS job queue in the QGPL library. The job is routed using the routing data APPLICS. If the job is an interactive job, the job queue must be allocated by an active subsystem.

Error messages for TFRJOB

***ESCAPE Messages**

CPF1289

Transfer job is not allowed.

CPF1315

Command &1 not allowed in this environment.

CPF1357

Job not transferred.

CPF1364

Job not transferred. Job queue &1 in library &2 not active.

CPF1365

Job not transferred. Subsystem &1 ending.

CPF1366

Subsystem &1 has no usable work station entry for &2.

CPF1367

User &1 not authorized to subsystem &2

CPF1368
&1 not authorized to job queue &2 in library &3.

CPF1369
Job queue &1 in &2 not found.

CPF1370
Job queue &1 in &2 not accessible.

CPF1372
Job not transferred. Job currently being ended.

CPF1373
Job not transferred. System request in effect for job.

CPF1375
Job not transferred. Single active device not allowed to transfer.

TFRPASTHR (Transfer Pass-Through) Command Description

TFRPASTHR Command syntax diagram

Purpose

The Transfer Pass-Through (TFRPASTHR) command is used to transfer from a pass-through system to a source system. The TFRPASTHR command performs the same function as a System Request (SYS REQ) option 10, 11, 13, or 14, and is valid only on a target pass-through system.

Optional Parameter

TOJOB
Specifies the program that is given control when the user is transferred to the home system or the previous system.

***SRC:** The job on the current system is suspended, and control is transferred back to the program specified on the SRQ10PGM parameter of the Start Pass-Through (STRPASTHR) command on the previous system. When the specified program ends, the target system gets control.

***ALT:** The job on the target system is suspended, and control is transferred back to the alternate job on the previous system. When control is transferred, the Transfer Job (TFRJOB) command can be used to transfer from the alternate job to the original job, giving control to the target system. Otherwise, when the alternate job ends, the target system gets control.

***HOME:** The job on the target system is suspended, and control is transferred back to the program specified on the SRQ10PGM parameter of the Start Pass-Through (STRPASTHR) command on the home system. When the specified program ends, the target system gets control.

***HOMEALT:** The job on the target system is suspended, and control is transferred back to the alternate job on the home system. When control is transferred, the Transfer Job (TFRJOB) command can be used to transfer from the alternate job to the original job, giving control to the target system. Otherwise, when the alternate job ends, the target system gets control.

Example for TFRPASTHR

```
TFRPASTHR  TOJOB(*HOME)
```

This command transfers control back to the source job on the home system.

No error messages.

TFRSECJOB (Transfer Secondary Job) Command Description

TFRSECJOB Command syntax diagram

Purpose

The Transfer Secondary Job (TFRSECJOB) command creates a secondary interactive job at your work station, then transfers control between the primary and secondary jobs. The first time you issue this command, you receive the signon prompt for the secondary job. Once you sign on, a secondary job is created, allowing you to receive the basic working display of the new job (such as the command entry display). Your primary job remains suspended as long as you remain in your secondary job. The next time you issue the TFRSECJOB command, your current job is suspended, and you return to the first job at the point at which you left it. When you sign off either job, you are automatically returned to the remaining job.

Note: This command performs the same function as option 1 on the System Request Menu.

There are no parameters for this command.

Example for TFRSECJOB

```
TFRSECJOB
```

The job that is currently running is suspended, and the SIGNON prompt is displayed. If a secondary job already exists, that job resumes running (the SIGNON prompt is not displayed).

Error messages for TFRSECJOB

***ESCAPE Messages**

CPF1380
Transfer to secondary interactive job not valid.

CPF1381
Transfer to secondary interactive job not valid.

CPF1383
Transfer to secondary interactive job not valid.

CPF1384
Transfer to a secondary interactive job not valid.

TFRGRPJOB (Transfer to Group Job) Command Description

TFRGRPJOB Command syntax diagram

Purpose

The Transfer to Group Job (TFRGRPJOB) command suspends the job that issued the TFRGRPJOB command and the group job specified by the GRPJOB parameter is resumed (if it already exists) or is created (if it does not exist). In both cases, control is transferred to the job specified by the GRPJOB parameter.

The job issuing the TFRGRPJOB command remains suspended until control is passed back to it and the job is resumed. A suspended job cannot do any processing until it is resumed. Suspended group jobs can regain control if another group job issues the TFRGRPJOB command to transfer to the suspended job, or if the active job in the group ends (and the suspended job was specified as the job to gain control).

If the work station message queue is in break or notify mode in the job that issued the TFRGRPJOB command, the message queue is set to the same mode in the group job that is transferred to. For example, if group job A has the work station message queue in break mode and transfers to group job B, the queue is in break mode in group job B. If group job B changes the queue to notify mode and transfers back to group job A, the queue is in notify mode in group job A.

If a user message queue is associated with the group (Change Group Attributes (CHGGRPA) command), it is handled like the work station message queue when one group job transfers to another group job.

Note: No message is sent for normal completion of the TFRGRPJOB command. Instead, the user may obtain a control code describing the circumstances under which the job regained control (along with the group job name and the job number of the previously active job) by using the RTVGRPA command. For more information about these control codes, see the Retrieve Group Attributes (RTVGRPA) command.

Optional Parameters

GRPJOB

Specifies the name of the group job to which control is transferred.

***PRV:** Control is transferred to the previously active job in the group. If the previously active job no longer exists, then the most recently active job in the group is resumed. This special value is valid only if there is another group job in the group.

***SELECT:** The group job selection display is shown. The user can choose which group job to transfer to or create a new group job and transfer to it.

group-job-name: Specify the group job name of the job to which control is transferred.

When a group job name is specified and that group job already exists, the group job is resumed. Whenever an already existing group job is resumed, the INLGRPPGM, SPCENV, and TEXT parameters are ignored. However, when a group job name is specified and that group job does not already exist, the INLGRPPGM parameter is required and must specify an existing program. When this is the case, a group job is started and control is passed to the new group job.

INLGRPPGM

Specifies the qualified name of the job's first group program. The INLGRPPGM parameter is valid only when a group job is created. If the group job being transferred to already exists, this parameter is ignored.

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.

initial-group-program-name: Specify the name of the first group program.

SPCENV

Specifies the environment in which the group job starts. This parameter is valid only when this command creates a group job. If control is transferring to an existing group job, this parameter is ignored.

***DFT:** The new group job starts in the environment specified in the user profile (the environment in which this command is running). The group job starts in the System/36 environment if one of the following is true:

- The System/36 environment is active in the job in which this command is running.
- The user profile specifies that the user runs in the System/36 environment, and the first program called in the group job is QCMD.

***INLGRPPGM:** The new group job starts in the environment determined by the first program called in the group job. If the first group program is QCMD, the special environment value in the user profile is used to determine the environment.

***S36:** The new group job starts in the System/36 environment.

***NONE:** The new group job starts in the iSeries 400 environment.

RSTDSP

Specifies whether data being shown at a display device by this display file is saved at the time the file is suspended (made temporarily inactive) so that a different display file can be used to show different data on the same device.

***NO:** The data being shown by this file is not saved when the file is suspended.

***YES:** The data being shown when the file is suspended is saved so it can be restored to the display of the device when the file is used again.

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

This text appears on the Group Job Selection display.

Note: This parameter only has meaning when a group job is created. If the group job being transferred to already exists, this parameter is ignored.

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

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

Example for TFRGRPJOB

```
TFRGRPJOB  GRPJOB(GROUPJ1)  INLGRPPGM(QGPL/PROGRAM1)
  SPCENV(*S36)
```

This command suspends running of the current job. If group job GROUPJ1 already exists, it is resumed at the point where it was suspended (the next high-level language command following the TFRGRPJOB request).

If group job GROUPJ1 does not exist, group job GROUPJ1 is created and runs the program QGPL/PROGRAM1 in the System/36 environment.

Error messages for TFRGRPJOB

***ESCAPE Messages**

CPF1E15
Problem occurred while calling Operational Assistant.

CPF1310
Request to transfer to group job failed with reason code &1.

CPF1313
Value &1 for parameter &2 not allowed name.

CPF1314
Value &1 for parameter &2 not allowed.

CPF9871
Error occurred while processing.

UPDPGM (Update Program) Command Description

UPDPGM Command syntax diagram

Purpose

The Update Program (UPDPGM) command can be used to replace modules of an Integrated Language Environment (ILE) bound program with other modules on the system, without requiring you to change or recompile the bound program. Modules being replaced must be module objects (*MODULE) on the system.

Other jobs running the bound program can run while the program is being updated with this command. The currently running bound program is moved to library QRPLOBJ and an updated version of the bound program is inserted into the library of the bound program. Current activations of the program will continue running with the QRPLOBJ version.

Restrictions:

1. You must have *CHANGE authority to the library of the bound program.
2. You must have *USE, *OBJMGT, and *OBJEXIST authorities to the bound program.
3. You must be the owner, a member of a group who is the owner of the bound program, or be a user with *ALLOBJ authority.
4. You must have *USE authority to the following:
   - *MODULE objects specified on the MODULE parameter, and their libraries.
   - *SRVPGM objects specified on the BNDSRVPGM parameter, and their libraries.
   - *BNDDIR objects specified on the BNDDIR parameter, and their libraries, and all objects used to resolve external symbols for these *BNDDIR objects, and their libraries.

Required Parameters

PGM Specifies the name of the bound program that is to be updated.

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

<u>*USRLIBL:</u> 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.

library-name: Specify the name of the library where the bound program is located.

program-name: Specify the name of the bound program that is to be updated.

MODULE

Specifies the names of the existing *MODULE objects that are to replace the modules of the same name in the bound program. If two or more modules of the bound program have the same name, the RPLLIB parameter indicates which is to be replaced.

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

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

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

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

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

***ALL:** All modules of the same name, to which the user has authority, replace the modules of the bound program.

***NONE:** No modules are specified.

generic-module-name:* Specify the generic name of the modules that replace the modules of the bound program. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All modules with names that begin with the generic name, and for which the user has authority, replace the modules of the bound program.

module-name: Specify the name of the module that replaces a module of the bound program.

Optional Parameters

RPLLIB

Specifies the method used to select the module to be replaced when two or more modules of the bound program have the name specified on the MODULE parameter.

***ONLY:** The bound program contains only one module of the specified name and it is replaced. If two or more modules of the bound program have the specified name, an exception is signaled and the bound program is not updated.

***FIRST:** The first module of the specified name in the module list of the bound program is replaced.

***MODULE:** The module that originated from the same library as the specified module is replaced. If no module of the specified name originally came from the same library as the replacing module, no module is replaced and an exception is signaled.

library-name: Specify the name of the originating library of the module to be selected for replacement. If no module of the specified name originated in the specified library, no module is replaced.

BNDSRVPGM

Specifies the service program to examine for exports if import requests to resolve external symbols cannot be met by the modules and service programs of the updated bound program. If the specified service program can resolve external symbols, it is added to the service programs that are bound to the bound program.

***NONE:** No service programs are examined during symbol resolution.

The specified service program can be qualified by one of the following 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. You cannot specify QTEMP for a library name.

***ALL:** All service programs are examined during symbol resolution.

generic-service-program-name:* Specify the generic name of the service program to examine during symbol resolution. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All service programs with names that begin with the generic name, and for which the user has authority, are examined during symbol resolution.

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

SRVPGMLIB

Library name to use to resolve to currently bound service programs. A value other than *SAME for this parameter may be specified if the program attribute ALWLIBUPD is *YES.

The possible library values are:

***SAME:** Use the library name where the service program (*SRVPGM) is currently bound from.

***LIBL:** All libraries in the job's library list will be searched until the first match is found for each bound *SRVPGM. The first occurrence of a *SRVPGM will be used to resolve to currently bound service programs and *LIBL is saved to be used at run time. If no match is found in the job's library list, the *SRVPGM currently bound to the program will be used. You must have *USE authority to the *SRVPGM objects in the library specified and *EXECUTE authority to the library itself. Note the service programs that came from the implicit binding directories (system-supplied service programs) will not be changed.

library-name: The name of the library to be used first to resolve to all currently bound service programs. If a bound *SRVPGM does exist in the library specified on SRVPGMLIB parameter, that *SRVPGM from that library will be used instead of the currently bound *SRVPGM and the library name specified on the SRVPGMLIB parameter is saved to be used at run time. If a *SRVPGM does not exist in the library specified on the SRVPGMLIB parameter, the *SRVPGM currently bound to the program will be used. You must have *USE authority to the *SRVPGM objects in the library specified and *EXECUTE authority to the library itself. Note the service programs that came from the implicit binding directories (system supplied service programs) will not be changed. You cannot specify QTEMP for a library name.

BNDDIR

Specifies the binding directory to examine for exports if import requests to resolve external symbols cannot be met by (1) the modules and service programs of the updated bound program or by (2) the service program specified on the BNDSRVPGM parameter. If a module or service program listed in the specified binding directory can resolve external symbols, it is added to the modules or service programs that are bound to the bound program.

***NONE:** No binding directories are specified.

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

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

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

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

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

binding-directory-name: Specify the name of the binding directory to be used during symbol resolution.

ACTGRP

Activation group name.

The possible values are:

***SAME:** The activation group will not be changed. Specify this value if the program was given *CALLER or *NEW activation group at the time it was created.

activation-group-name: The name of the activation group that is associated with this called program. If the program was given a named activation group at the time it was created, the name of that activation group can be changed to another named activation group. Note that changing the activation group name can affect the behavior of the program (or service program)". Please refer to the ILE Concepts book for detailed information on the behavior of named activation groups.

OPTION

Specifies the options to be used when the bound program is updated.

Program Creation Objects

***GEN:** An updated program object is created.

***NOGEN:** An updated program object is not created.

Duplicate Procedure Name Options

***DUPPROC:** During symbol resolution, the procedures that are exported from the modules and service programs need not be unique. The first procedure of the specified module and service programs that matches the import request is exported.

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

Duplicate Variable Name Options

***DUPVAR:** During symbol resolution, the variables that are exported from the modules and service programs need not be unique. The first variable of the specified modules and service programs that matches the import request is exported.

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

Diagnostic Message Options

***WARN:** The appropriate diagnostic messages are signaled. Also, if you specify duplicate procedures or variables (*DUPPROC or *DUPVAR) and duplicates are found, a diagnostic message is issued indicating what duplicates were found.

***NOWARN:** No information or diagnostic messages are issued.

Trimming Marooned Modules Options

A **marooned module** is a module of the bound program being updated. This module was originally bound into the bound program from a binding directory to resolve one or more imports. Imports are not resolved for this program update.

***NOTRIM:** Marooned modules are not removed from the bound program.

Note: Programs may grow significantly over time when *NOTRIM is specified.

***TRIM:** Marooned modules are removed from the bound program.

Note: If marooned modules are removed from the bound program for this program update, the exports that the modules contain are not available for other program updates. System objects referred to by the removed module are not removed (see the DSPPGMREF command).

Resolving References Options

***RSLVREF:** All imports must be resolved to exports for the bound program to be updated.

***UNRSLVREF:** All imports do not need to resolve to exports for the bound program to be updated.

Note: If this command contains an import that does not resolve, an exception will be generated when the command is run.

DETAIL

Specifies the level of detail of the binder listing to be printed. The printer file *LIBL/QSYSPRT is used to create the listing.

***NONE:** No binder listing is printed.

***BASIC:** The brief summary table, the options passed to this command, and some processing statistics are printed.

***EXTENDED:** The extended summary table and the binding information listing are printed, in addition to the information provided in the *BASIC listing (the brief summary table, the options passed to this command, and some processing statistics).

***FULL:** The Cross-reference listing is printed, in addition to the information provided in the *EXTENDED listing (the extended summary table, the binding information listing, the brief summary table, the options passed to this command, and some processing statistics).

Example for UPDPGM

```
UPDPGM   PGM(STAR)  MODULE(SKY/NOVA)  RPLLIB(*FIRST)
```

This command replaces the first module named NOVA existing in the program object STAR with the module NOVA in the library SKY.

Error messages for UPDPGM

*ESCAPE Messages

CPF223D
Not authorized to update &1 in &2 type *&3.

CPF223E
Authority check for use adopted authority attribute failed.

CPF5CA7
SRVPGMLIB must be *SAME when ALWLIBUPD is *NO.

CPF5CE0
Program &1 not updated.

CPF5CE2
Unexpected error occurred during program or service program update.

CPF5D1B
Update of program &1 in library &2 not allowed.

UPDSRVPGM (Update Service Program) Command Description

UPDSRVPGM Command syntax diagram

Purpose

The Update Service Program (UPDSRVPGM) command can be used to replace modules of an Integrated Language Environment (ILE) bound service program with other modules on the system, without requiring you to change or recompile the bound service program. Modules being replaced must be module objects (*MODULE) on the system.

Other jobs running the bound service program can run while the service program is being updated with this command. The currently running service program is moved to library QRPLOBJ and an updated version of the service program will be inserted into the library of the service program. Current activations of the service program will continue running with the QRPLOBJ version.

Restrictions:

1. You must have *CHANGE authority to the library of the bound service program.
2. You must have *USE, *OBJMGT, and *OBJEXIST authorities to the bound service program.
3. You must be the owner, a member of a group who is the owner of the bound service program, or be a user with *ALLOBJ authority.
4. You must have *USE authority to the following:
   - *MODULE objects specified on the MODULE parameter, and their libraries.
   - *SRVPGM objects specified on the BNDSRVPGM parameter, and their libraries.
   - *BNDDIR objects specified on the BNDDIR parameter, and their libraries, and all objects used to resolve external symbols for these *BNDDIR objects, and their libraries.

Required Parameters

SRVPGM
Specifies the name of the bound service program that is to be updated.

The name of the bound service 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.

library-name: Specify the name of the library where the bound service program is located.

service-program-name: Specify the name of the bound service program that is to be updated.

MODULE

Specifies the names of the existing *MODULE objects that are to replace the modules of the same name in the bound program. If two or more modules of the bound program have the same name, the RPLLIB parameter indicates which is to be replaced.

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

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

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

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

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

***ALL:** All modules of the same name to which the user has authority replace the modules of the bound service program.

***NONE:** No modules are specified.

Note: Specifying *NONE for this parameter allows updating of other service program attributes. The existing modules(s) in the service program are used with the other parameters specified on the UPDSRVPGM command.

generic-module-name:* Specify the generic name of the modules that replace the modules of the bound program. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All modules with names that begin with the generic name, and for which the user has authority, replace the modules of the bound service program.

module-name: Specify the name of the module that replaces a module of the bound service program.

Optional Parameters

RPLLIB

Specifies the method used to select the module to be replaced when two or more modules of the bound program have the name specified on the MODULE parameter.

***ONLY:** The bound service program contains only one module of the specified name and it is replaced. If two or more modules of the bound service program have the specified name, an exception is signaled and the bound service program is not updated.

***FIRST:** The first module of the specified name in the module list of the bound service program is replaced.

***MODULE:** The module that originated from the same library as the specified module is replaced. If no module of the specified name originally came from the same library as the replacing module, no module is replaced and an exception is signaled.

library-name: Specify the name of the originating library of the module to be selected for replacement. If no module of the specified name originated in the specified library, no module is replaced.

EXPORT

Specifies the variables and procedures that are to be exported from the updated service program. This parameter also specifies whether new signatures identifying the sequence of exports in the service program are created.

***CURRENT:** The variables, procedures, and signatures currently exported from the service program continue to be exported. No new signatures are created.

Note: If a variable or procedure that is currently exported is not available for export after the update, an exception is signaled and the service program is not updated.

***SRCFILE:** The variables and procedures of the source file and source member specified on the SRCFILE and SRCMBR parameters are exported. If the specified source file differs from the one used to create the service program, a new signature or set of signatures may be created.

Note: If a signature is lost, some current clients of the service program may not be able to use the service program without binding again.

***ALL:** All variables and procedures exported from the modules of the service program are exported from the updated service program.

If the number or names of the variables and procedures exported before the service program update differs from those exported after the service program update, a new signature is created.

Note: If a new signature is created, all clients of the service program must be bound again before they can use the service program.

SRCFILE

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

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

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

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

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

QSRVSRC: The default source file containing the specifications for exporting variables and procedures is used.

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

SRCMBR

Specifies the name of the member in the source file specified on the SRCFILE parameter that contains the specifications for exporting variables and procedures from this bound service program.

***SRVPGM:** The source file member has the same name as the service program being updated.

source-file-member-name: Specify the name of the member that contains the specifications for exporting.

BNDSRVPGM

Specifies the service program to examine for exports if import requests to resolve external symbols cannot be met by the modules and service programs of the updated bound service program. If the specified service program can resolve external symbols, it is added to the service programs that are bound to the bound service program.

***NONE:** No service programs, except those in the bound service program being updated, are examined during symbol resolution.

The specified service program can be qualified by one of the following 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. You cannot specify QTEMP for a library name.

***ALL:** All service programs are examined during symbol resolution.

generic-service-program-name:* Specify the generic name of the service program to examine during symbol resolution. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All service programs with names that begin with the generic name, and for which the user has authority, are examined during symbol resolution.

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

SRVPGMLIB

Library name to use to resolve to currently bound service programs. A value other than *SAME for this parameter may be specified if the service program attribute ALWLIBUPD is *YES.

The possible library values are:

***SAME:** Use the library name where the service program (*SRVPGM) is currently bound from.

***LIBL:** All libraries in the job's library list will be searched until the first match is found for each bound *SRVPGM. The first occurrence of a *SRVPGM will be used to resolve to currently bound service programs and *LIBL is saved to be used at run time. If no match is found in the job's library list, the *SRVPGM currently bound to the service program will be used. You must have *USE authority to the *SRVPGM objects in the library specified and *EXECUTE authority to the library itself. Note the service programs that came from the implicit binding directories (system-supplied service programs) will not be changed.

library-name: The name of the library to be used first to resolve to all currently bound service programs. If a bound *SRVPGM does exist in the library specified on SRVPGMLIB parameter, that *SRVPGM from that library will be used instead of the currently bound *SRVPGM and the library name specified on the SRVPGMLIB parameter is saved to be used at run time. If a *SRVPGM does not exist in the library specified on the SRVPGMLIB parameter, the *SRVPGM currently bound to the service program will be used. You must have *USE authority to the *SRVPGM objects in the library specified and *EXECUTE authority to the library itself. Note the service programs that came from the implicit binding directories (system supplied service programs) will not be changed. You cannot specify QTEMP for a library name.

BNDDIR

Specifies the binding directory to examine for exports if import requests to resolve external symbols cannot be met by (1) the modules and service programs of the updated bound program or by (2) the service program specified on the BNDSRVPGM parameter. If a module or service program listed in the specified binding directory can resolve external symbols, it is added to the modules or service programs that are bound to the bound service program.

***NONE:** No binding directories, except those in the bound service program being updated, are examined during symbol resolution.

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

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

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

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

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

binding-directory-name: Specify the name of the binding directory to be used during symbol resolution.

ACTGRP

Activation group name.

The possible values are:

***SAME:** The activation group will not be changed. Specify this value if the service program was given *CALLER activation group at the time it was created.

activation-group-name: The name of the activation group that is associated with this called service program. If the service program was given a named activation group at the time it was created, the name of that activation group can be changed to another named activation group. Note that changing the activation group name can affect the behavior of the program (or service program)".

Please refer to the ILE Concepts book for detailed information on the behavior of named activation groups.

OPTION

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

Program Creation Options

***GEN:** An updated service program object is created.

***NOGEN:** An updated service program object is not created.

Duplicate Procedure Name Options

***DUPPROC:** During symbol resolution, the procedures that are exported from the modules and service programs need not be unique. The first procedure of the specified modules and service programs that matches the import request is exported.

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

Duplicate Variable Name Options

***DUPVAR:** During symbol resolution, the variables that are exported from the modules and service programs need not be unique. The first variable of the specified modules and programs that matches the import request is exported.

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

Diagnostic Message Options

***WARN:** The appropriate diagnostic messages are signaled. Also, if you specify duplicate procedures or variables (*DUPPROC or *DUPVAR) and duplicates are found, a diagnostic message is issued indicating what duplicates were found.

***NOWARN:** No information or diagnostic messages are issued.

Trimming Marooned Modules Options

A **marooned module** is a module of the bound service program being updated. This module was originally bound into the bound service program from a binding directory to resolve one or more imports. Imports are not resolved for this service program update.

***NOTRIM:** Marooned modules are not removed from the bound service program.

Note: Bound service programs may grow significantly over time when *NOTRIM is specified.

***TRIM:** Marooned modules are removed from the bound service program.

Note: If marooned modules are removed from the bound program during this program update, the exports that the modules contain are not available for other service program updates. System objects referred to by the removed module are not removed (see the DSPPGMREF command).

Resolving References Options

***RSLVREF:** All imports must resolve to exports for the bound service program to be updated.

***UNRSLVREF:** All imports do not need to resolve to exports for the bound service program to be updated.

Note: If this command contains an import that does not resolve, an exception will be generated when the command is run.

DETAIL

Specifies the level of detail of the binder listing to be printed. The printer file *LIBL/QSYSPRT is used to create the listing.

***NONE:** No binder listing is printed.

***BASIC:** The brief summary table, the options passed to this command, and some processing statistics are printed.

***EXTENDED:** The extended summary table and the binding information listing are printed, in addition to the information provided in the *BASIC listing (the brief summary table, the options passed to this command, and some processing statistics).

***FULL:** The cross-reference listing is printed, in addition to the information provided in the *EXTENDED listing (the extended summary table, the binding information listing, the brief summary table, the options passed to this command, and some processing statistics).

Example for UPDSRVPGM

```
UPDSRVPGM   SRVPGM(WORKDOC)  MODULE(BIN/TASKTWO)
  RPLLIB(*MODULE)
```

This command replaces the module named TASKTWO in the service program object named WORKDOC with another module named TASKTWO in the library BIN, only if the module being replaced was originally from the library BIN.

Error messages for UPDSRVPGM

***ESCAPE Messages**

CPF223D
Not authorized to update &1 in &2 type *&3.

CPF223E
Authority check for use adopted authority attribute failed.

CPF5CA7
SRVPGMLIB must be *SAME when ALWLIBUPD is *NO.

CPF5CE1
Service program &1 not updated.

CPF5CE2
Unexpected error occurred during program or service program update.

CPF5D1C
Update of service program &1 in library &2 not allowed.

UPDSYSINF (Update System Information) Command Description

UPDSYSINF Command syntax diagram

Purpose

The Update System Information (UPDSYSINF) command updates the following information on the system that was retrieved through the use of the RTVSYSINF command:

1. Edit descriptions
2. Network attributes
3. Reply list entries
4. Service attributes
5. Service providers
6. System values

Required Parameters

LIB Specifies the library where the system information exists.

library-name: Specify the library name where the system information exists.

Optional Parameters

TYPE Specifies the type of system information being updated on the system.

***ALL:** All information is updated.

***EDTD:** The edit descriptions that were saved are updated.

***NETA:** The network attributes that were saved are updated.

***RPYLE:** The reply list entries that were saved are updated.

***SRVA:** The service attributes that were saved are updated.

***SRVPVD:** The service providers that were saved are updated.

***SYSVAL:** The system values that were saved are updated.

Example for UPDSYSINF

```
UPDSYSINF  LIB(TEST)  TYPE(*ALL)
```

This command will update all saved system information on the current system from the information in library TEST.

Error messages for UPDSYSINF

***ESCAPE Messages**

CPFA976
Error occurred updating system information for type &1.

VRYCFG (Vary Configuration) Command Description

VRYCFG Command syntax diagram

Purpose

The Vary Configuration (VRYCFG) command varies configuration objects on or off and optionally resets the input/output processor (IOP) associated with the specified objects. The VRYCFG command is used to vary on or off one or more configuration objects with the capability of also varying on the downline attached configuration objects. The configuration object types that can be varied on or off are network server, network interface, line, controller, and device. This command applies to all such objects on the system.

For the configuration object type of media library resource, this command can be used to reset the drives within a tape media library device or change the allocation of drives within a tape media library device or an optical media library device. To determine the current allocation of drive resources, use the Work with Media Library Status (WRKMLBSTS) command.

Downline attached objects can be varied on or off along with the specified object by specifying the value *NET on the range parameter (RANGE(*NET)).

The range parameter is ignored for network server objects. A vary on or off of a network server varies the network server and the attached lines. Varying on multiple network servers in parallel may be done by specifying a generic name or list of names for the CFGOBJ parameter and the value *YES for the submit multiple jobs parameter (SBMMLTJOB(*YES)). If using the submit multiple jobs function, specify a batch subsystem description that is configured to allow enough jobs to simultaneously run to optimize this operation. The job will be submitted to run under the current user profile of the job doing the VRYCFG.

External LAN TCP/IP interfaces attached to network server objects of type *WINDOWSNT are automatically started by default and can optionally be controlled with the start TCP/IP interfaces parameter.

Downline attached objects of a network interface description are all the attached lines and all controllers attached to the lines and all the devices attached to the controllers. Downline attached objects of a line are all the attached controllers and all the devices attached to the controllers. Downline attached objects of a controller are all the attached devices. Devices do not have downline attachments. The RANGE parameter has no effect when varying devices.

Varying on network servers, network interfaces, and lines synchronously or asynchronously can be controlled by the VRYWAIT parameter. For lines, this applies only to Ethernet, token-ring, X.25, or switched IDLC, SDLC, BSC, and Async line descriptions. The value specified for the VRYWAIT parameter determines how long the system waits until either the object is varied on before completing the VRYCFG command, or until the timer expires.

The time required to vary on an object is the time it takes to:

- Put tasks in place to manage the resource
- Activate the communications input/output processor (IOP)
- Download the IOP microcode
- Establish communications with the data circuit-terminating equipment (DCE) and other unique protocol set up tasks

Line vary on time does not include telephone dialing time, however; a powered off modem may prevent vary on completion and cause the wait time to expire. If the timer expires, an informational message is sent to the QSYSOPR message queue. This is followed by the vary completion message.

The VRYCFG command can also be used to reset the IOP. An IOP can be a communications controller, local work station, or magnetic media controller. An IOP reset is valid only when varying on network interface descriptions, lines (except twinaxial data link controller (TDLC) lines), local work station controllers, tapes, and diskettes.

When varying on a network server, an IOP reset is always done. The RESET parameter is ignored for network servers.

A line cannot be varied on:

- For frame relay and IDLC lines, until the network interface description is varied on
- For switched lines, the vary does not complete until a dial connection has been completed
- If it is attached to a network server. Vary on the network server description, which varies on the attached lines.

A controller cannot be varied on:

- For leased (nonswitched) lines, if the line to which it is attached is varied off
- For switched lines, until a dial connection has been completed

A device cannot be varied on:

- If the controller to which it is attached is varied off. This does not apply to tape and diskette devices because they are not attached to a controller.

A network server cannot be varied off:

- Until all attached devices and controllers are varied off. Varying off the server also varies off the attached line descriptions.

- If any iSeries 400 clients have files open on the server

Note: Use the Work with Network Server Status (WRKNWSSTS) command (available from Work with Configuration Status display) to determine the status of network server sessions with other clients.

A network interface description cannot be varied off:

- Until all attached lines, controllers, and devices are varied off

A line cannot be varied off:

- Until all the attached controllers and devices are varied off
- If it is attached to a network server. Vary off the network server to vary off the attached lines.

A controller cannot be varied off:

- If it is being used or allocated for use
- Until all the attached devices are varied off

A device cannot be varied off:

- If it is being used or allocated for use

Required Parameters

CFGOBJ

Specifies the name of the configuration object to be varied on or off. The name can have up to 10 characters.

***ANYNW:** All controller descriptions that speccify a link type of *ANYNW will be varied on or off. This value is only valid if CFGTYPE is *CTL.

***APPN:** All objects that use Advanced Peer-to-Peer Networking (APPN) will be varied on or off. This value is only valid if CFGTYPE is *CTL or *DEV.

***PRVCFGTYPE:** Process all objects that were processed the last time this command was run in this job for the specified configuration object type.

generic-configuration-description-name:* Specify the generic name of the configuration description 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.

Notes:

1. If no generic names are specified, objects are processed in the order the names were entered for this parameter.
2. If one or more generic names are specified, all objects matching the name(s) specified will be processed in alphabetical order rather than in the order the names were entered.
3. Generic names can not be specified if the CFGTYPE parameter is *MLBRSC.
4. Care must be taken when varying objects using generic names or special values so that unintentional processing of objects does not occur. For example, using the generic name CTL* will cause all objects beginning with the letters CTL to be processed even though the user may have only intended some of these objects to be processed.

CFGTYPE

Specifies the type of configuration object to have its status changed.

***NWS:** The network server and attached lines are varied on or off.

***NWI:** The network interface is varied on or off.

***LIN:** The line is varied on or off.

***CTL:** The controller is varied on or off.

***DEV:** The device is varied on or off.

***MLBRSC:** The status for drives within a media library is changed.

STATUS

Specifies the status to which the configuration object is changing.

***ON:** The object is varied on.

***OFF:** The object is varied off.

***RESET:** The drive resources of the tape media library device are reset.

Note: The drive resources must be specified on the RSRCNAME parameter.

The media library device must be varied on before this value can be specified.

***ALLOCATE:** For a tape media library device, the drive resources are allocated for use only by this system. If the library device is shared by multiple systems, other systems cannot use these drives while this device description is varied on.

For an optical media library device, the drive within the optical media library is allocated for use. This drive is only available for use by this media library resource. The optical library resource must be varied on to perform this action.

Note: The drive resources must be specified on the RSRCNAME parameter.

***UNPROTECTED:** The drive resources of the tape media library device can be used by all systems that share this library device.

***DEALLOCATE:** For a tape media library device, the drive resources are deallocated for this system. If the tape media library is shared by multiple systems, the drives cannot be used by this system but can be used by other systems.

For an optical media library device, the drive within the optical media library is no longer available for use. The drive cannot be used by another library resource. The optical library resource must be varied on to perform this action.

Note: The drive resources must be specified on the RSRCNAME parameter.

Optional Parameters

RANGE

Specifies which configuration elements are varied on or off.

Notes:

1. For device descriptions: Because devices do not have downline attached objects, value *NET is not valid.

2. For switched line descriptions: When varying on elements, value *NET is not valid. When varying off elements and specifying value *NET, the line and its downline attached objects are varied off.
3. For network interface descriptions: Specifying value *NET varies on all nonswitched attachments and varies off all nonswitched attachments.
4. For network server descriptions, this parameter is ignored. Varying on or off a network server also varies the attached lines.

***NET:** All downline attached configuration elements are varied on or off.

***OBJ:** Only the specified objects are varied on or off.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For a synchronous vary on, this parameter specifies how long the system waits for the vary on to complete.

Notes:

1. If the VRYWAIT parameter is specified on the VRYCFG command for a line description that is not Ethernet, token-ring, DDI, X.25, or switched SDLC, BSC, or Async, the parameter is accepted but ignored.
2. If the VRYWAIT parameter is specified on the VRYCFG command for a server description, the parameter is accepted but ignored.

***CFGOBJ:** The VRYWAIT parameter value specified in the line description, network interface description, or server description is used.

***NOWAIT:** The system does not wait for vary on completion. The line description, network interface description, or network server description is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. When ONLINE(*YES) is used, specifying a wait time in the line description affects system IPL time. In such cases, system IPL time is influenced by the amount of time required to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to:
   - Put tasks in place to manage the line
   - Activate the communications I/O processor (IOP), including downloading the IOP model-unique Licensed Internal Code
   - Establish the communications tasks and processes

   Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

ASCVRYOFF

Specifies whether the vary off is asynchronous. This parameter is not allowed when STATUS(*ON) is specified.

***NO:** The vary off is synchronous.

***YES:** The vary off is asynchronous.

RESET

Specifies whether the IOP associated with the object is reset.

Note: Network server descriptions are always reset when varied on; the RESET parameter is accepted but ignored.

***NO:** The associated IOP is not reset.

***YES:** The associated IOP is reset.

RSRCNAME
Specifies the resource name of the drive within the media library device to be reset or reallocated.

RESETCFGF
Specifies whether to reset the configuration file that is associated with a *BASE or *NETWARE network server description. If there is no configuration file associated with *BASE or *NETWARE network server description, this parameter is ignored. This parameter is valid only when CFGTYPE is *NWS.

***NO:** The configuration file is not reset.

***YES:** The configuration file is reset.

FRCVRYOFF
Specifies whether inquiry messages for active jobs will be suppressed. This parameter is not allowed when STATUS(*ON) is specified.

***NO:** Inquiry messages will be presented for active jobs.

***YES:** Inquiry messages will be suppressed for active jobs and the jobs will be ended.

***LOCK:** For devices other than APPC and Intra, an attempt will be made to get a lock on the device description no matter what its current status might be. If the lock is successfully obtained, it will be transferred to the system job assigned to hold the device description lock when the device is in a varied off state. If the device is in a state other than varied off, an attempt to vary off the device description will also be made.

STRTCPIFC
Specifies whether or not to start the TCP/IP interfaces associated with the external LAN ports 1 and 2 of a network server description of type *WINDOWSNT. This parameter is ignored if the network server description that is being varied on is not of type *WINDOWSNT. This parameter is valid only when CFGTYPE is *NWS.

***YES:** The external LAN TCP/IP interfaces associated with ports 1 and 2 are started.

***NO:** The external LAN TCP/IP interfaces associated with ports 1 and 2 are not started.

SBMMLTJOB

» Specifies whether the network server or auxilliary storage pool (ASP) vary operation is done as multiple separate jobs or as part of this job. «

***NO:** The vary operation is done in the current job.

***YES:**

» A specific job is submitted to vary each network server or auxilliary storage pool (ASP) device. The job attributes and job queue are determined from the job description (JOBD) parameter. The job uses the user's profile. «

JOBD Specifies the job description for the job that is being submitted for the vary operation. This parameter may only be specified if the submit multiple jobs (SBMMLTJOB) parameter is *YES.

QBATCH: The job description QBATCH is to be used for the job.

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.

Examples for VRYCFG

Example 1: Varying On the Network Interface and Downline Attachments

```
VRYCFG   CFGOBJ(NWI1)  CFGTYPE(*NWI)  STATUS(*ON)
```

This command varies on the network interface and all downline attachments.

Example 2: Varying Off the Line and Attached Downline Objects

```
VRYCFG   CFGOBJ(LINE1)  CFGTYPE(*LIN)  STATUS(*OFF)
```

This command varies off the line and all attached downline objects. The RANGE parameter took the default value of *NET.

Example 3: Varying on the Controller

```
VRYCFG   CFGOBJ(CONTROLLER1)  CFGTYPE(*CTL)  STATUS(*ON)
  RANGE(*OBJ)
```

This command varies on only the controller.

Example 4: Varying on the Device

```
VRYCFG   CFGOBJ(DEVICE1)  CFGTYPE(*DEV)  STATUS(*ON)
  RANGE(*NET)
```

This command varies on only the device. Note the RANGE parameter value has no effect on devices.

Example 5: Varying on the Line and Resetting the IOP

```
VRYCFG   CFGOBJ(LINE1)  CFGTYPE(*LIN)  STATUS(*ON)
  RANGE(*OBJ)  RESET(*YES)
```

This command varies on only the line and resets the associated IOP.

Example 6: Using Line Description Value for Wait Time

```
VRYCFG   CFGOBJ(LINE1)  OJBTYPE(*LIN)  STATUS(*ON)
  RANGE(*OBJ)  VRYWAIT(*CFGOBJ)
```

This command varies on only the line and uses the vary wait time value specified in the line description for LINE1.

Example 7: Using 80 Seconds as Vary Wait Time

```
VRYCFG   CFGOBJ(LINE1)  CFGTYPE(*LIN)  STATUS(*ON)
  RANGE(*OBJ)  VRYWAIT(80)
```

This command varies on only the line using 80 seconds as the vary wait time value.

Example 8: Varying on a Network Server

```
VRYCFG   CFGOBJ(SERVER1)  CFGTYPE(*NWS)  STATUS(*ON)
```

This command varies on the network server description named SERVER1 and its attached line descriptions. The vary on wait value specified in the network server description is used. Note that the RANGE and RESET parameters are ignored for network servers if they are specified.

Example 9: Resetting Drives Within a Media Library

```
VRYCFG   CFGOBJ(MYLIBRARY)  CFGTYPE(*MLBRSC)
  STATUS(*RESET)  RSRCNAME(TAP01 TAP02)
```

This command resets the drives TAP01 and TAP02 within the media library device MYLIBRARY. The device MYLIBRARY must be varied on to perform this action.

Example 10: Deallocating Drives Within a Media Library

```
VRYCFG   CFGOBJ(MYLIBRARY)  CFGTYPE(*MLBRSC)
  STATUS(*DEALLOCATE)  RSRCNAME(OPT02)
```

This command deallocates drive OPT02 within the media library device MYLIBRARY. The device MYLIBRARY must be varied on to perform this action.

Example 11: Varying On Multiple Network Servers in Parallel

```
VRYCFG   CFGOBJ(IPCS*)  CFGTYPE(*NWS)
  STATUS(*ON)  SBMMLTJOB(*YES) JOBD(*LIBL/QBATCH)
```

This command submits a separate batch job to perform the vary on for each network server which has a name that begins with IPCS. The number of jobs that run in parallel depends on the configuration of the subsystem being used.

Error messages for VRYCFG

***ESCAPE Messages**

CPF26AF
Status of drive resources in device description &1 not changed.

CPF26B6
Initialization program has ended with a hard error.

CPF26B7
Initialization program ended with soft error.

CPF262E
Vary on at IPL processing stopped due to failure.

CPF262F
QDCTRF stopped due to failure.

CPF2640
Vary command not processed.

CPF2659
Vary command may not have completed.

APING (Verify APPC Connection) Command Description

APING Command syntax diagram

For the description of the APING command, see the VFYAPPCCNN (Verify APPC Connection) command description.

VFYAPPCCNN (Verify APPC Connection) Command Description

VFYAPPCCNN Command syntax diagram

Purpose

The Verify APPC Connection (VFYAPPCCNN) command, also known as APING, exchanges data packets between the local location and the specified remote location using Advanced Program-to-Program Communications (APPC), and measures the round-trip time of each data exchange.

For this function to work, the remote location specified must be running the target portion of this function, **APINGD** (APING daemon).

Required Parameters

RMTLOCNAME
Specifies the remote location to connect with. Specify the remote location name using the format nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name. If only the remote location name is specified, the local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID).

MODE Specifies the APPC mode that is to be used.

***NETATR:** The mode in the network attributes is used.

mode-name: Specify a mode name. Specify BLANK for a mode name consisting of eight blank characters.

Note: SNASVCMG and CPSVCMG are reserved names and cannot be specified.

RMTUSER
Specifies the user identifier (ID) for the target system. If a user ID is specified for this parameter and password security is active on the target system, RMTPWD(*NONE) is not valid.

***NONE:** No user ID is sent. If security on the target system is configured to require a user ID, this command will fail.

***CURRENT:** The user ID of the job (signed-on user) using this command is sent.

remote-user-identifier: Specify a user ID to use that exists on the target system. If a user ID is specified and password security is active on the target system, a password must be specified.

RMTPWD
Specifies the password sent to the target system.

***NONE:** The system does not pass a password. If a user identifier (ID) is specified on the RMTUSER parameter and password security is active on the target system, this command will fail.

password: Specify a password sent to the target system to verify the sign-on of the user ID specified in the RMTUSER parameter. The password may or may not be substituted across the communication line depending on whether the remote system supports password substitution.

MSGMODE
Specifies the amount of information displayed by the command.

***VERBOSE:** Display verification message after each iteration.

***QUIET:** Display only the initial APING message and the summary message.

PKTLEN
Specifies the length (in bytes) of the packets that are exchanged between the local and remote systems.

100: The packet length is 100 bytes.

packet-length: Specify the length of the packet. Valid values range from 0 through 32763 bytes.

NBRITER
Specifies the number of iterations. For each iteration, the specified number of data packets are exchanged between the local and remote systems, and a timing measurement is taken.

2: Two iterations are performed.

number-of-iterations: Specify the number of iterations. Valid values range from 1 through 32767.

NBRPKT
Specifies the number of packets that are sent by the local system for each iteration before giving the target system permission to send.

1: One packet is sent for each iteration.

number-of-packets: Specify the number of packets that are sent for each iteration. Valid values range from 1 through 32767.

ECHO Whether the remote location should echo packets back to the local location.

YES: Packets are echoed from the remote location back to the local location.

***NO:** Packets are sent from the local location to the remote location only; packets are not echoed back to the local location.

WAITTIME
Specifies the time in seconds to wait for the return (echo) before declaring the remote location to be unreachable.

10: The system waits 10 seconds for the remote location to respond.

***NOMAX:** The system waits indefinitely.

***NOWAIT:** The system returns immediately if there is not a connection ready and available.

wait-time: Specify the wait time in seconds. Valid values are 2 to 3600 seconds.

Examples for VFYAPPCCNN

Example 1: Verify APPC Connection

```
VFYAPPCCNN   RMTLOCNAME(RPCNET.CHICAGO)
  NBRITER(3)  NBRPKT(4)  PKTLEN(500)
```

This command exchanges four 500-byte packets in each of three iterations with remote location CHICAGO, network identifier RPCNET. The default mode used is taken from network attribute DFTMODE. Since the default MSGMODE(*VERBOSE) was taken, each iteration will result in an informational message in the job log indicating the elapsed time for the iteration.

Example 2: APING

```
APING   RMTLOCNAME(RPCNET.CHICAGO)
  NBRITER(3)  NBRPKT(4)  PKTLEN(500)
```

This command is equivalent to the command in example 1.

Example 3: APING

```
APING RMTLOCNAME(RPCNET.CHICAGO)
      WAITTIME(20)
```

This command verify the connection with remote location CHICAGO, network identifier RPCNET. The maximum time to wait for a response from the remote location is 20 seconds.

Error messages for VFYAPPCCNN

***ESCAPE Messages**

CPF91CC
Command did not complete successfully.

VFYCMN (Verify Communications) Command Description

VFYCMN Command syntax diagram

Purpose

The Verify Communications (VFYCMN) command shows the Select a Line to Test display, which can be used to verify that communications equipment is operating properly.

Depending on the user's system configuration, the following tests can be run:

- Link
- Local modem
- Remote modem
- Cable
- Communications input/output adapter
- Link Problem Determination Aid-2 (LPDA-2)

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

VFYTYPE
Specifies whether local or remote communications hardware is checked to verify that it is operating properly.

Note: The System Manager for iSeries 400 licensed program must be installed to do remote analysis.

***LOCAL:** Communications hardware is checked to verify that it is operating properly on the local iSeries 400.

***REMOTE:** Communications hardware is checked to verify that it is operating properly on another iSeries 400 that is enrolled as a service requester.

RCPNAME
Specifies the remote control point name for the service requester system where the remote verification is done.

Note: This parameter is valid only when VFYTYPE(*REMOTE) is specified.

NETID Specifies the network identifier (ID) of the service requester system where the remote verification is done.

Note: This parameter is valid only when VFYTYPE(*REMOTE) is specified.

***NETATR:** The network ID of the service provider is used.

network-identifier: Specify the network ID of the service requester system where the remote verification is done.

USERID
Specifies the user identifier (ID) used to access the remote system.

Note: This parameter is valid only when VFYTYPE(*REMOTE) is specified.

PASSWORD
Specifies the password used to access the remote system.

Note: This parameter is valid only when VFYTYPE(*REMOTE) is specified.

***NONE:** No password is needed to access the remote system because the remote system has a security level of 10.

password: Specify the password needed to access the remote system.

Examples for VFYCMN

Example 1: Showing Select a Line to Test Display

```
VFYCMN
```

This command shows the Select a Line to Test display.

Example 2: Checking a Remote System

```
VFYCMN  VFYTYPE(*REMOTE)
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values, remote analysis begins.

Example 3: Accessing a Remote System Using a Password

```
VFYCMN  VFYTYPE(*REMOTE)  RCPNAME(RCH38377)
  USERID(JON)  PASSWORD
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values beyond the ones specified on the command example, remote analysis begins.

Example 4: Accessing a Remote System Without a Password

```
VFYCMN  VFYTYPE(*REMOTE)  RCPNAME(RCH38377)
  USERID(JON)
```

This command is similar to the preceding example except that the PASSWORD parameter is not specified. The same prompt display is shown, however, the system assumes that the remote system has a security level of 10, that is, it does not use passwords. Another prompt display appears after this command is specified. After the user specifies the appropriate values on this display, remote analysis begins.

Example 5: Checking a Local System

```
VFYCMN  VFYTYPE(*LOCAL)
```

This command begins analysis on the local device. The remaining parameters do not appear on the display.

Error messages for VFYCMN

***ESCAPE Messages**

CPF2B3C
Licensed program &1 not installed.

»

VFYIMGCLG (Verify Image Catalog) Command Description

VFYIMGCLG Command syntax diagram

Purpose

The Verify Image Catalog (VFYIMGCLG) command is used to verify the images in an image catalog based on the value specified in the TYPE parameter. The user can optionally sort the images in install sequence based also on the TYPE parameter.

A status message will be issued upon successful completion of the command. If the VFYIMGCLG command fails, the Work with Image Catalog Entries (WRKIMGCLGE) command can be used to look at the images and the status of each. The VFYIMGCLG command is intended for verifying images for a complete software upgrade, licensed program install, installation of a PTF, or other types of installs.

Restriction:

1. You must have *SECADM and *ALLOBJ special authorities to use this command.

Required Parameter

IMGCLG
Specifies the name of the image catalog to be verified.

image-catalog-name: Specify the name of the image catalog to be verified.

Optional Parameters

TYPE Specifies the type of image catalog to be verified.

***UPGRADE:** The image catalog to be verified is for a complete software upgrade. The system will verify that the necessary images for a software upgrade exist and can be loaded into the virtual optical device.

***PTF:** The image catalog to be verified is for a PTF install. The system will verify all PTF volume sets are complete and can be loaded into the virtual optical device. All non-PTF volumes will be unloaded.

***OTHER:** The image catalog to be verified is not for a specific type of install. This option will load the images from the image catalog in the order they exist. There will be no verification or sorting of images.

SORT Specifies whether the images of this type should be sorted in the order required for a software upgrade or PTF install. If TYPE(*OTHER) is specified, the images in the image catalog are not sorted.

***NO:** The images in the image catalog are not sorted based on the value specified for the TYPE parameter.

***YES:** The images in the image catalog are sorted based on the value specified for the TYPE parameter.

Examples for VFYIMGCLG

Example 1: Verify Image Catalog for TYPE(*UPGRADE)

```
VFYIMGCLG  IMGCLG(MYCLG)  TYPE(*UPGRADE)  SORT(*YES)
```

This command verifies that the install image catalog MYCLG contains the necessary files for a software upgrade. If the necessary media files exist, the images will be sorted in the order required for a software upgrade.

Example 2: Verify Image Catalog for TYPE(*PTF)

```
VFYIMGCLG  IMGCLG(MYCLG)  TYPE(*PTF)
```

This command verifies that all required cumulative PTF volumes in image catalog **MYCLG** are available. No sorting of the images will occur.

Error messages for VFYIMGCLG

***ESCAPE Messages**

CPFBC20
Verify for image catalog &1 failed.

CPFBC40
Not authorized to command &1.

CPFBC41
&1 command failed.

***STATUS Messages**

CPCBC20
Image catalog &1, type &2 verified.

«

VFYLNKLPDA (Verify Link Supporting LPDA-2) Command Description

VFYLNKLPDA Command syntax diagram

Purpose

The Verify Link Supporting LPDA-2 (VFYLNKLPDA) command allows the user to run a Link Problem Determination Aid-2 (LPDA-2) test and then receive the results in a format specified by the user.

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

Required Parameter

LINE Specifies the name of the line of the link to be tested.

Optional Parameters

TEST Specifies which test to run.

***DCELINSTS:** Checks the status of the line and data circuit-terminating equipment (DCE).

***DCELINTST:** Tests the line and the data circuit-terminating equipment (DCE).

***ANZLIN:** Analyzes the line. This special value is valid only for analog lines.

***SNDRCV:** Tests sending and receiving capabilities.

OUTPUT

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

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

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

LCLDCEADR

Specifies the hexadecimal address of a local DCE. By convention, bits 4 through 7 of this byte indicate the Link Segment Level (LSL) of the local DCE and the remote DCE. Bits 0 through 3 are used to uniquely define a local DCE among several DCEs on the same LSL. The address is set on the local DCE during configuration and must follow this convention.

Note: '00'X is not a valid address for a local DCE.

***LCL:** '01'X, the address for the local DCE on LSL 1, is used.

local-DCE-address: Specify the address of the local DCE. Valid values range from X'01' through X'FB'.

RMTDCEADR

Specifies the hexadecimal address of a remote DCE.

This parameter must be specified if the user is testing a multipoint link.

***ANY:** 'FD'X, the global address of a remote DCE, is used. If a remote DCE is not idle, it responds to the 'FD'X address, regardless of its previously configured address.

Note: Multipoint tributary DCEs will not respond to an address of *ANY.

remote-DCE-address: Specify the address of the remote DCE. Valid values range from X'01' through X'FB'.

SEQCOUNT

Specifies the number of sequences to transmit during a send and receive test. A sequence is a group of 16 blocks. The block length is determined depending on how a DCE is configured.

1: One sequence is transmitted.

number-of-sequences: Specify the number of sequences to transmit. Valid values range from 1 through 3.

DTEPORT

Specifies the data terminal equipment (DTE) port on the remote DCE. The DTE status for this port is returned to the user. This parameter is valid only when working with line status and line testing of DCEs. In addition, this parameter is only useful for multiport DCEs.

A: The A-port is used.

B: The B-port is used.

C: The C-port is used.

D: The D-port is used.

DTERTY

Specifies whether the user is attempting to retry a test request from the system DTE to a local DCE due to a bad response or no response received from a local DCE.

***NO:** This is not a DTE retry.

***YES:** This is a DTE retry.

DCERTY

Specifies that the local DCE should retry a link operation to a remote DCE if a bad response or no response is received from the remote DCE.

***NO:** A DCE retry is not performed.

***YES:** A DCE retry is performed, if necessary.

VRYLNKSTS

Specifies whether the link is varied on or varied off following a completed test.

After running a test on a manually switched link, the link should be left varied on to allow further information to be received on the same connection. If a switched link is varied off, the connection may fail and no further analysis can be performed.

***SAME:** The value does not change.

***ON:** The link remains varied on after testing has completed.

***OFF:** The link is varied off after testing has completed.

Examples for VFYLNKLPDA

Example 1: Checking Line Status

```
VFYLNKLPDA  LINE(LINE1)  DTEPORT(B)
```

This command retrieves the DCE line status from synchronous data link control (SDLC) line, LINE1, and displays the status. The remote DCEs DTE line connection status of port B is returned if the user is verifying a multiport DCE. An error message will be returned if the remote DTE has only a single port, for example, port A. The default VRYLNKSTS(*SAME) causes the line named LINE1 to return to the status prior to the test.

Example 2: Analyzing a Line

```
VFYLNKLPDA   LINE(LINE2)  TEST(*ANZLIN)
  OUTPUT(*PRINT)  LCLDCEADR(02)  VRYLNKSTS(*ON)
```

This command analyzes the SDLC line, LINE2. The second LSL is used; the lower four bits of the local DCE address (LCLDCEADR) are 2. The results are sent to a spooled file. After the test, LINE2 remains varied on to allow for more testing.

Example 3: Testing Sending and Receiving Capabilities

```
VFYLNKLPDA   LINE(LINE3)  TEST(*SNDRCV)
  SEQCOUNT(3)  RMTDCEADR(21)  DCERTY(*YES)
```

This command tests the sending and receiving capabilities on the multipoint line, LINE3. Three sequences of 16 blocks are sent between the local (control) DCE and the remote (tributary) DCE with the address of X'21'. If the local DCE fails to receive a response on the first attempt, the local DCE will retry this command to the remote DCE.

Error messages for VFYLNKLPDA

*ESCAPE Messages

CPF1BAF
Error occurred while processing VFYLNKLPDA command.

CPF1BA9
Line &1 vary off failed.

CPF1BCC
Test cannot be run at this time.

CPF1BCD
DCE self test failed.

CPF1BCE
Sense byte returned is not valid.

CPF1BC1
Error detected while processing VFYLNKLPDA command.

CPF1BC3
Test cannot run in switched network backup mode.

CPF1BC4
Requested test is not supported.

CPF1BC5
Required feature not installed.

CPF1BC6
Required feature not operational.

CPF1BC7
Test is not compatible with DCE configuration.

CPF1BC8
DTEPORT parameter cannot be specified.

CPF1BD1
Line description &1 is not *SDLC.

CPF1BD2
System Service Tools is active.

CPF1BD4
Not authorized to line description &1.

CPF1BD7
VFYLNKLPDA command does not support switched lines.

CPF1B8A
Line &1 failed during test.

CPF1B8B
No response received for test request.

CPF1B8C
Test cannot be run on line &1.

CPF1B8D
Error occurred while processing VFYLNKLPDA command.

CPF1B8E
Test cannot be run at this time.

CPF1B8F
Test request failed. Test already active on line.

CPF1B80
Line description &1 does not exist.

CPF1B81
Error occurred while getting configuration information.

CPF1B83
Line &1 is not in proper state for test.

CPF1B89
Test cannot be run on line &1.

CPF1B9F
Line &1 cannot be varied off at this time.

CPF1B93
Line &1 did not vary on.

VFYMOVBRM (Verify Moves 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.

VFYMOVBRM Command syntax diagram

Purpose

The Verify Moves using BRM (VFYMOVBRM) command shows you the Verify Media Moves display. The Verify Media Moves display allows you to verify individual volume movement, verify all volumes scheduled to move or cancel movement for individual or selected volumes.

The Volume Movement Report is produced when you run the VFYMOVBRM command. The report is written to printer file QP1AVMS.

There are no parameters for this command.

Error messages for VFYMOVBRM

None

VFYNTWAUTE (Verify NetWare Authentication Entry) Command Description

VFYNTWAUTE Command syntax diagram

Purpose

The Verify NetWare Authentication Entry (VFYNTWAUTE) command verifies the authentication entry for a server. The user name, password, and other data, are sent to the server, where they are used to sign on to the server. This command could be used, for example, to verify that the password is correct before submitting a batch job that uses the server. Use the Work with NetWare Authentication Entries (WRKNTWAUTE) command to view the server authentication entries in a user profile.

Note: This command cannot be used for entries that specify PASSWORD(*STRNTWCNN).

Required Parameter

SVRTYPE
Specifies the type of the server authentication entry that is to be verified.

***NETWARE3:** The entry is for a NetWare 3.x server.

***NDS:** The entry is for a NetWare Directory Services tree.

Optional Parameters

NDSTREE
For *NDS entries, specifies the NDS tree name of the authentication entry to be verified.

SERVER
For *NETWARE3 servers, specifies the server name of the authentication entry to be verified.

'server-name': Use the specified server name.

USRPRF
Specifies the user profile containing the authentication entry.

***CURRENT:** Use the current user profile.

user-profile-name: Use the specified user profile. The user profile must be the current user profile, or the user must have *USE and *OBJMGT authority to the user profile, and *SECADM special authority.

Example for VFYNTWAUTE

```
VFYNTWAUTE SVRTYPE(*NETWARE3) SERVER(SERVER03)
```

This command verifies the server authentication entry for NetWare 3.x server SERVER03 from the current user profile.

Error messages for VFYNTWAUTE

***ESCAPE Messages**

FPE021A
Verification of authentication entry failed.

VFYOPT (Verify Optical) Command Description

VFYOPT Command syntax diagram

Purpose

The Verify Optical (VFYOPT) command verifies whether a specified optical drive unit or a specified optical media library unit is operating.

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

DEV Specifies the name of the optical drive or the optical media library unit whose operation is being verified.

Examples for VFYOPT

Example 1: Verifying an Optical Drive

```
VFYOPT   DEV(OPT1)
```

This command verifies whether the optical drive unit named OPT1 is operating.

Example 2: Verifying an Optical Media Library

```
VFYOPT   DEV(OPTMLB1)
```

This command verifies whether the optical media library unit named OPTMLB1 is operating.

Error messages for VFYOPT

***ESCAPE Messages**

CPF2C31
Optical unit description &1 not found.

CPF2C33
Device description &1 not an optical unit.

VFYOPCCNN (Verify OptiConnect Connections) Command Description

VFYOPCCNN Command syntax diagram

Purpose

The Verify OptiConnect Connections (VFYOPCCNN) command verifies that all OptiConnect connections are operational.

There are no parameters for this command.

Examples for VFYOPCCNN

```
VFYOPCCNN
```

This command verifies connections with all other systems that are connected to the requesting system through OptiConnect.

No error messages.

VFYPRT (Verify Printer) Command Description

VFYPRT Command syntax diagram

Purpose

The Verify Printer (VFYPRT) command runs the 3287, 3812 SCS, 3812 IPDS, 4210, 4214, 4224, 4234, 4245, 5219, 5256, 5262, 5524, 5525, 5553, and 5583 Printers by causing them to print a test pattern a specified number of times.

Restrictions:

1. This command does not support advanced printer function printers that are configured with AFP(*YES) specified in their printer device descriptions. (Note that some advanced function printers, such as the 3820, 3827, and 3855 Printers, can be configured only in this manner.) This means that this command *can* exercise both *nonadvanced* function printers and advanced function printers, such as the 3812 and 3816 Printers, that have AFP(*NO) specified in their device descriptions.
2. This command is shipped with public *EXCLUDE authority.
3. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSRV
   - QSRVBAS

Required Parameters

DEV Specifies the name of the printer on which to run the test pattern. The device name must be the same as that specified in the device description for the printer.

Optional Parameters

TIMES

Specifies the number of times that the specified printer prints the test pattern.

1: The test pattern is printed once.

number-of-times: Specify a value for the number of times to print the test pattern.

Example for VFYPRT

```
VFYPRT   DEV(PRTR3)  TIMES(15)
```

This command causes printer PRTR3 to print a test pattern 15 times.

Error messages for VFYPRT

***ESCAPE Messages**

CPF3943

Incorrect value specified for device parameter.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

CPF9831

Cannot assign device &1.

CPF9845

Error occurred while opening file &1.

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

VFYTAP (Verify Tape) Command Description

VFYTAP Command syntax diagram

Purpose

The Verify Tape (VFYTAP) command verifies whether a specified tape unit is operating.

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

Required Parameter

DEV Specifies the name of the tape unit whose operation is being verified.

***RSRCNAME:** The resource name of the tape unit whose operation is being verified is used.

device-name: Specify the name of the tape unit whose operation is being verified.

Optional Parameter

RSRCNAME
Specifies the resource name of the tape unit whose operation is being verified.

Example for VFYTAP

```
VFYTAP  DEV(TAP3)
```

This command verifies whether the tape unit named TAP3 is working.

Error messages for VFYTAP

***ESCAPE Messages**

CPF2B31
Tape unit description &1 not found.

CPF2B32
Resource &1 not found.

CPF2B33
Device description &1 not a tape unit.

CPF2B34
Resource &1 not a tape unit.

CPF2B35
Tape verification not available for ’&1’ type tape units.

CPF2B36
No device description was found for resource &1.

CPF2B37
Tape verification request not correct.

CPF2B39
Problem analysis did not complete due to an error.

VFYTCPCNN (Verify TCP/IP Connection) Command Description

VFYTCPCNN Command syntax diagram

Purpose

The Verify TCP/IP Connection (VFYTCPCNN) command, also known as PING, tests the Transmission Control Protocol/Internet Protocol (TCP/IP) connection between a system and the remote system specified on the remote system parameter.

Notes:

- The VFYTCPCNN (PING) command cannot be used to verify IP over SNA connections.
- The local domain name is used by many applications including PING. PING appends the local domain to a host name if a domain is not specified or if a period (.) does not appear at the end of the specified host name.

Required Parameter

RMTSYS

Specifies the remote system name of the host with which the Verify TCP/IP operation takes place. To be successful, the name must be valid, and the remote system must be able to communicate with the local system. You can assign names to an internet address by using either of the following:

- Work with Host Table menu, which is an option on the Configure TCP/IP menu.
- Remote name server to map a remote system name to an internet address.

» Host name resolution will depend on the value specified for the ADRVERFMT parameter. «

***INTNETADR:** » The remote system is identified by the address specified for the INTNETADR parameter. «

remote-system: Specify the remote system name to be verified.

Optional Parameters

INTNETADR

Specifies the remote internet address. » Either a valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 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.

» **ADRVERFMT**

Specifies which IP address version protocol (IP Version 4 or IP Version 6) is used to resolve host names specified by RMTSYS.

***CALC:** The address family will be ’calculated’ (determined) based on the address entered by the RMTSYS parameter. VFYTCPCNN (PING) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution will then be used in order to resolve the IP address.

***IP4:** Explicitly use the IP Version 4 protocol addressing method.

***IP6:** Explicitly use the IP Version 6 protocol addressing method. «

MSGMODE

Specifies the amount of information displayed by the VFYTCPCNN command.

ELEMENT 1: RESPONSE MESSAGE DETAIL

Specifies how much information is displayed during each VFYTCPCNN response.

***VERBOSE:** Display messages as each PING response arrives.

***QUIET:** Display only the initial VFYTCPCNN message and the summary messages.

» ELEMENT 2: SUMMARY, IF RESPONSE ERRORS

Specifies the type of summary message returned if VFYTCPCNN fails.

***COMP:** If the VFYTCPCNN request fails then the summary message is sent as a completion message.

***ESCAPE:** If the VFYTCPCNN request fails then the summary is sent as a monitorable escape message. This is useful if you have written a program to issue the VFYTCPCNN request and wish to monitor the VFYTCPCNN request for errors. See the error message section of the VFYTCPCNN command help, for a list of possible escape messages. «

PKTLEN

Specifies the length (in bytes) of the packets that are sent to the remote system.

256: The packet length is 256 bytes.

packet-length: Specify the length of the packet. Valid values range from 8 through 512 bytes.

NBRPKT

Specifies the number of packets that are sent to the remote system.

5: Five packets are sent.

number-of-packets: Specify the number of packets that are sent to the remote system. Valid values range from 1 through 999.

WAITTIME

Specifies the time in seconds to wait for the return (echo) packet before declaring this packet transfer a failure.

1: The system waits 1 second.

time-to-wait-for-reply: Specify the length of time in seconds. Valid values range from 1 through 600 seconds.

LCLINTNETA

Specifies the local internet address of the interface that the outbound packets are to use. » Either a valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 internet adderess 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.

***ANY:** Use any interface's local internet address.

local-ip-address: Specify the local internet address to be verified.

TOS »

Note: This parameter is not used if IP Version 6 address resolution is used for connection to the remote system.

«

Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

***NORMAL**: Normal service is used for delivery of data.

***MINDELAY**: Minimize delay means that prompt delivery is important for data on this connection.

***MAXTHRPUT**: Maximize throughput means that a high data rate is important for data on this connection.

***MAXRLB**: Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

***MINCOST**: Minimize monetary cost means that lower cost is important for data on this connection.

Note: If you issue VFYTCPCNN using a local internet address (LCLINTNETA) to specify an interface to be used for outbound packets, you must also specify a type of service (TOS) that matches that interface.

IPTTL Specifies the IP time-to-live value. The IP datagram time-to-live value specifies a relative limit on the number of hops across which an IP datagram remains active. The time-to-live value acts as a “hop count” that is decremented by each gateway to prevent internet routing loops.

»

Note: IP Version 6 refers to this parameter as the “hop limit”.

«

***DFT:** Use the default IP time-to-live value. For multicast addresses the default value is 1. The default value is the IPTTL value defined in Change TCP/IP Attributes (CHGTCPA).

ip-time-to-live: Specify an IP time-to-live value. Valid values range from 1 through 255.

Examples for VFYTCPCNN

Example 1: Verify TCP/IP Connection with a specified host name

```
VFYTCPCNN RMTSYS(IPHOST) PKTLEN(100) NBRPKT(10)
  WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system (known to the TCP/IP configuration as IPHOST) over a TCP/IP link. Each packet transfer must take place within 15 seconds or it fails.

Example 2: Verify TCP/IP Connection with an IP address and with some defaults changed

```
VFYTCPCNN RMTSYS(*INTNETADR) INTNETADR('128.1.1.10')
  PKTLEN(100) NBRPKT(10)  WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system over a TCP/IP interface. The user represents the RMTSYS with its internet address 128.1.1.10, rather than with an assigned system name. Each packet transfer that takes more that 15 seconds fails.

Example 3: Verify TCP/IP Connection with a specified host name and using a specific local interface address

```
VFYTCPCNN RMTSYS(IPHOST) MSGMODE(*QUIET)
  LCLINTNETA('9.2.2.3')
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system over a specific TCP/IP interface that has the local address 9.2.2.3.

Because MSGMODE(*QUIET) is specified, only the primary output messages are displayed. The interface parameter is useful on multi-homed hosts to verify network connectivity through a specific physical interface.

Example 4: Verify TCP/IP Connection with an IP Version 6 address

```
VFYTCPCNN RMTSYS(*INTNETADR) INTNETADR('1:2:3:4:5:6:7:8')
```

This command attempts to verify the TCP/IP connection of a remote system that has the local address of 1:2:3:4:5:6:7:8.

Example 5: Verify TCP/IP Connection with a specified IP Version 6 defined host name

```
VFYTCPCNN RMTSYS(IPV6HOST)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

The default “Address version format” is *CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to verify its connection over a TCP/IP link.

Example 6: Verify TCP/IP Connection with a specified IP Version 6 defined host name and explicitly use IP Version 6 host name resolution

```
VFYTCPCNN RMTSYS(IPV6HOST) ADRVERFMT(*IP6)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

This example differs from example 5 in that only a valid IP version 6 resolved address, for IPV6HOST, will be used when attempting to verify its connection over a TCP/IP link.

No error messages.

WAIT (Wait) Command Description

WAIT Command syntax diagram

Purpose

The Wait (WAIT) command accepts input from any display device from which user data is requested by one or more previous Receive File (RCVF), Send File (SNDF), or Send/Receive File (SNDRCVF) commands that do not wait to receive the input data. Those commands had *NO specified in the WAIT parameter or, in the case of SNDF, had the INVITE DDS keyword option specified in the record format sent to the display, and specified a particular device file to receive and transfer the data to the CL program. Only one input request per device can be outstanding at any given time. If there are multiple outstanding input requests, the user data of the first device to respond to the specified device file is sent to the CL program. If the data is received within the wait interval, the Wait operation ends and the next command in the program is processed. Otherwise an escape message is sent to the CL program.

The program waits the number of seconds specified on the WAITRCD keyword of the Create Display File (CRTDSPF), Change Display File (CHGDSPF), or Override with Display File (OVRDSPF) commands for a device to respond to an input request.

Restriction: This command is valid only for display files within a CL program. It cannot be used with database files.

Optional Parameter

DEV Specifies the name of the CL variable that receives the name of the display device that responds with user data for the CL program.

***NONE:** No CL variable name is specified; the name of the responding device is not needed.

CL-variable-name: Specify the name (up to 10 characters) of the CL variable that receives the name of the responding device. A device name cannot be specified in this parameter.

Examples for WAIT

Example 1: Receiving User Data

```
DCLF   FILE(MSCREEN)
  *
  *
  *
RCVF   DEV(DEV1)  WAIT(*NO)
  *
  *
  *
RCVF   DEV(DEV2)  WAIT(*NO)
  *
  *
  *
WAIT  DEV(&DEVNAM)
```

In this example, the device file MSCREEN is used to receive user data. The RCVF commands specify that the program does not wait for the data. The WAIT command causes the program to wait for the display device file MSCREEN to pass input data to it from one of its devices. The name of the responding display device is placed in the CL variable &DEVNAM. The received data is placed in the program variables associated with the record format of the declared file.

Example 2: Receiving Data in the Program Variable

```
DCLF   FILE(DF1)
  *
  *
  *
RCVF   DEV(DEV1)  WAIT(*NO)
  *
  *
  *
RCVF   DEV(DEV2)  WAIT(*NO)
  *
  *
  *
WAIT   DEV(&DN)
```

In this example, the RCVF commands specify that the program does not wait for user data. When the WAIT command is issued, the name of the responding device is placed in the CL variable &DN. The data received from the device file DF1 is placed in the program variable of the associated record format in the file DF1.

Error messages for WAIT

*ESCAPE Messages

CPF3204
Cannot find object needed for file &1 in &2.

CPF7306
Member &1 not added to file &2 in &3.

WRKACTJOB (Work with Active Jobs) Command Description

WRKACTJOB Command syntax diagram

Purpose

The Work with Active Jobs (WRKACTJOB) command allows the user to work with performance and status information for the active jobs in the system. The SEQ criteria can be changed by operations on the display. Selection values (CPUPCTLMT, SBS, RSPLMT, JOB) can also be changed by operations on the display.

Optional Parameters

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.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

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

RESET

Specifies whether the active job statistics are reset.

***NO:** The active job statistics are not reset. The measurement time interval is extended (similar to pressing the F5 key) if a previous display active jobs command has run in the current job. All active jobs are displayed.

***YES:** The active job statistics are reset. A measurement time interval of zero (0) is used (similar to pressing the F13 key). All active jobs are displayed.

SBS Specifies the names of the subsystems (or all subsystems) whose active jobs are displayed.

***ALL:** The active jobs in all subsystems are displayed. System jobs that are not associated with any subsystem are also displayed.

subsystem-name: Specify the name of the subsystem to be displayed. All active jobs in this subsystem (including the monitor) are displayed. Up to 25 subsystem names can be specified.

CPUPCTLMT

Specifies the minimum processing time percent value that a job must have before it is included on the display.

***NONE:** There is no minimum processing time limit that a job must have to be displayed.

CPU-percent-limit: Specify the minimum processing time percent limit that a job must have to be included on the display. Valid values range from 0.1 through 99.9.

RSPLMT

Specifies the minimum response time limit that a job must have before it is included on the display.

***NONE:** There is no minimum response time limit that a job must have to be displayed.

response-time-limit: Specify the minimum response time limit that a job must have to be included on the display. Valid values range from 0.1 through 999.9.

SEQ Specifies the sequence of the active jobs that are shown.

***SBS:** The jobs are ordered on the basis of the subsystem in which they are running. Jobs that run in a subsystem (auto-start jobs, interactive jobs, batch jobs, readers, and writers) are put in alphabetical order by job name, and are indented under the subsystem with which they are associated. Subsystem monitors (with the jobs in the subsystem grouped under each monitor job) are put in alphabetical order and presented before system (SYS) jobs. The system jobs (start operating system, system arbiter, and LU services) are put in alphabetical order by job name, and are presented after the subsystem monitors and jobs in the subsystems.

***JOB:** Jobs are put in alphabetical order by job name.

***TYPE:** Jobs are put in alphabetical order by job type and job name within the same type.

***POOL:** Jobs are ordered by the system pool in which they are running. The lowest values are presented first.

***PTY:** Jobs are ordered by priority of running. The highest priority values (0) are presented first.

***CPU:** Jobs are ordered by the amount of processing time they have used since the job started. The largest values are presented first.

***INT:** Jobs are ordered by the number of operator interactions that have occurred during the measurement interval. The largest values are presented first. Jobs that have no interactions (blank interaction field) come after jobs with 0 interactions.

***RSP:** Jobs are ordered by the average response time during the measurement interval. The largest values are presented first. Jobs that have no interactions (blank interaction field) come after jobs with 0 response time.

***AUXIO:** Jobs are ordered by the number of auxiliary storage input/output (I/O) operations that have occurred during the measurement time interval. The largest values are presented first.

***CPUPCT:** Jobs are ordered by the percent of CPU resource they have used during the measurement interval. The largest values are presented first.

***USER:** Jobs are ordered alphabetically by user name.

***FUNCTION:** Jobs are put in alphabetical order by the contents of the function field.

***STS:** Jobs are put in alphabetical order by the contents of the status field.

***NUMBER:** Jobs are ordered by job number. The largest values are presented first.

***THREADS:** Jobs are ordered by the number of active threads. The jobs with the largest number of active threads are presented first.

JOB Specifies the name of the active jobs to be displayed. Only active jobs within selected subsystems (based on the SBS parameter) are displayed. Subsystem monitor names only appear when *ALL or *SBS is specified. System jobs only appear when *ALL or *SYS is specified.

***ALL:** All the active jobs are displayed.

***SYS:** All active system jobs are displayed. If a value other than the default is specified in the SBS parameter when using this value, an error message is issued.

***SBS:** All active sybsystem monitors are displayed.

generic:* Specify all active jobs, that meet the criteria, that are to be displayed. System jobs and subsystem moniters are not displayed using this parameter.

job-name: Specify the active job that is to be displayed. System jobs and subsystem monitors are not displayed using this parameter.

INTERVAL

Specifies the interval (in seconds) to wait during the automatic refresh option. The default time is 300 seconds (5 minutes). Valid values range from 5 to 999 seconds. If this value is changed by the user, the value is saved and used as the default value. When automatic refresh is started the screen is refreshed automatically based on the time specified.

***PRV:** The interval of time used in the previous invocation. Until an interval is specified, 300 seconds is used.

automatic-refresh-interval: Specify the delay time (in seconds) for automatic refresh. Valid values range from 5 through 999.

Examples for WRKACTJOB

Example 1: Resetting Active Job Statistics

```
WRKACTJOB  RESET(*YES)  CPUPCTLMT(2)
```

This command allows the user to work with a display with no jobs appearing; the active job statistics are reset and no job has used any processing unit time since the reset point. When the display appears, the F5 key may be pressed; this causes a display of all jobs that have exceeded 2 percent of the processing unit utilization since the reset point.

Example 2: Working With Jobs in a Subsystem

```
WRKACTJOB  SBS(QINTER)  SEQ(*INT)
```

This command allows the user to work with all jobs in the QINTER subsystem. The sequence of the jobs is by the number of operator interactions, with the job with the most interactions appearing first.

Error messages for WRKACTJOB

***ESCAPE Messages**

CPF1093
Override of file device type not valid.

CPF9845
Error occurred while opening file &1.

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

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

CPF9850
Override of printer file &1 not allowed.

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

CPF9871
Error occurred while processing.

WRKALRD (Work with Alert Descriptions) Command Description

WRKALRD Command syntax diagram

Purpose

The Work with Alert Descriptions (WRKALRD) command allows the user to show, add, change, and remove alert description records. More information on alerts is in the Alerts Support book.

Optional Parameters

MSGID
Specifies the message ID at which the Work with Alert Descriptions display starts.

***FIRST:** The display begins with the first message identifier description found in the alert table.

message-ID: Specify the message identifier.

ALRTBL

Specifies the qualified name of the alert table.

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

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

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

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

QCPFMSG: The alert table named QCPFMSG is used.

alert-table-name: Specify the name of the alert table.

Example for WRKALRD

```
WRKALRD  MSGID(USR1234)  ALRTBL(USER/USRMSGS)
```

This command shows the Work with Alert Descriptions display, starting with message identifier USR1234 from alert table USRMSGS in library USER.

Error messages for WRKALRD

***ESCAPE Messages**

CPF2499
Message identifier &1 not allowed.

CPF7D41
Error occurred while logging order assistance request.

CPF7D42
Error occurred while performing database operation.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9811
Program &1 in library &2 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9821
Not authorized to program &1 in library &2.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9831
Cannot assign device &1.

CPF9871
Error occurred while processing.

WRKALRTBL (Work with Alert Tables) Command Description

WRKALRTBL Command syntax diagram

Purpose

The Work with Alert Tables (WRKALRTBL) command allows you to display and work with a list of alert tables, to change and delete specified alert tables, to work with the alert descriptions contained in specified alert tables, and to create new alert tables. More information on the alerts is in the Alerts Support book.

Restrictions:

1. Only the libraries to which you have USE authority will be searched.
2. Only the alert tables to which you have some authority will be shown on the display.
3. To perform operations on the alert tables, you must have USE authority to the command used by the operation, and the appropriate authority to the alert tables on which the operation is to be performed.

Required Parameter

ALRTBL
Specifies the qualified name of the alert table list that is shown.

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

***LIBL:** All libraries in the job’s library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

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

***ALL:** All the alert tables specified in the library are listed.

alert-table-name: Specify the name of the alert table that is shown.

generic-alert-table-name:* Specify the generic name of the alert table. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

Example for WRKALRTBL

```
WRKALRTBL   ALRTBL(ALRTBLLIB/AL*)
```

This command shows a list of all alert tables in library ALRTBLLIB whose names begin with AL. From the list shown, you can change, delete, or work with the alert descriptions in any or all of the alert tables shown. You can also create a new alert table.

Error messages for WRKALRTBL

***ESCAPE Messages**

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

WRKALR (Work with Alerts) Command Description

WRKALR Command syntax diagram

Purpose

The Work with Alerts (WRKALR) command allows the user to work with alerts that were created by the user's system or received from another system.

More information on alerts is in the Alerts Support book.

Optional Parameters

DSPOPT
Specifies whether alerts received from other systems and/or created locally are shown. Alerts that cannot be sent to the system focal point and are marked held can be shown.

***ALL:** All alerts that are received and locally created are shown.

***RCV:** Only alerts received from other systems are shown.

***LOCAL:** Only locally created alerts are shown.

***HELD:** All alerts that cannot be sent to the system's focal point and are marked HELD are shown.

Note: There is a distinction between held alerts that are sent or forwarded by this system, and held alerts that are received by another system. DSPOPT(*HELD) shows only held alerts that could not be sent or forwarded by this system.

PERIOD
Specifies the period of time for which the logged data is shown. The following values can be coded in this parameter, which contains two lists of two elements each. If PERIOD is not specified, the following values are assumed:

```
PERIOD((*AVAIL *BEGIN)
    (*AVAIL *END))
```

Element 1: Starting Time

One of the following is used to specify the time at or after which the data must have been logged to be shown. Any entries logged before the specified time and date are not shown.

***AVAIL:** The logged data that is available for the specified starting *date* is shown.

start-time: Specify the starting time for the specified starting date that indicates the logged data that is shown. 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 is used to specify the starting date on which or after which the data must have been logged. Any entries logged before the specified date are not shown.

***BEGIN:** The logged data from the beginning of the log is shown.

Note: If *BEGIN is specified, then any time value other than *AVAIL for start-time is ignored.

***CURRENT:** The logged data for the current day and between the specified starting and ending times (if specified) is shown.

start-date: Specify the date for which logged data is shown. The date must be entered in the format specified by the system values QDATFMT and, if separators are used, QDATSEP.

Element 3: Ending Time

One of the following is used to specify the ending time before which the data must have been logged.

***AVAIL:** The logged 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 logged data that is shown. See the description of *start-time* for details about how time can be specified.

Element 4: Ending Date

One of the following is used to specify the ending date before which or on which the data must have been logged.

***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 logged data is to be shown. The date must be entered in the format specified by the system values QDATFMT and, if separators are used, QDATSEP.

ALRTYPE

Specifies which types of alerts are shown. The alert type indicates the severity of the alert.

***ALL:** All types of alerts are shown.

Up to five of the following types of alerts can be specified.

***TEMP:** Alerts reporting a temporary problem are shown.

***PERM:** Alerts reporting a permanent problem are shown.

***PERF:** Alerts reporting a performance problem are shown.

***IMPEND:** Alerts reporting an impending problem are shown.

***PAFF:** Alerts reporting a problem about a permanently affected resource are shown. These alerts indicate that their originator has determined that the target resource is lost because of a persistent error in a resource other than the target.

***UNKNOWN:** Alerts reporting a problem with unknown severity are shown.

alert-type-code-point: Specify the code point for the alert type. The code point is specified by two (2) hexadecimal digits.

ALRRSC

Specifies the resource name of alerts that are reporting problems. Up to 50 resource names can be specified.

***ALL:** Alerts about failing resources are shown.

resource-name: Alerts that are reporting problems associated with the assigned resource name are shown.

ALRRSCTYPE

Specifies the resource types of alerts that are reporting problems. Up to 50 resource types can be specified. Each resource name has a resource type associated with that resource. Resource types are diskette (DKT) or tape (TAP), for example.

***ALL:** Alerts for all resource types are shown.

resource-type: Alerts that are reporting problems associated with the assigned resource type are shown.

ASNUSER

Specifies the user to which the alerts being shown are assigned. This value is taken from the value on the ASNUSER parameter in the Add Alert Action Entry (ADDALRACNE) command.

***ALL:** All alerts are shown.

***NONE:** The alerts not assigned to a user are shown.

user-name: Specify the name of the user to which the alerts being shown are assigned.

GROUP

Specifies the group to which the alerts being shown are assigned. This value is taken from the value on the GROUP parameter in the Add Alert Selection Entry (ADDALRSLTE) command.

***ALL:** All alerts are shown.

***DEFAULT:** The alerts assigned to the default group are shown.

***NONE:** The alerts not assigned to a group are shown.

group-name: Specify the name of the group to which the alerts being shown are assigned.

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.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

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

DETAIL

Specifies the level of detail contained in the printed listing. This parameter is valid only if OUTPUT(*PRINT) is specified.

***BASIC:** The output is a list of the alert resource, alert type, date, time, problem ID, assigned user, group assigned, alert description, and probable cause. The information on this list is only for alerts that meet requirements set by the subsetting parameters.

***EXTENDED:** The output is a spool file containing the WRKALR screen information, alert hierarchy, probable cause and recommended action, and details from the main detail panel. The spool file contains information only for the alerts that meet the user criteria.

***FULL:** The output is a spool file containing the same data as the *EXTENDED value output, plus additional information from the detail menu screens. Only the panels that contain data are printed.

Example for WRKALR

```
WRKALR  DSPORT(*LOCAL)  ALRTYPE(*TEMP *PERM)
  ALRRSCTYPE(DKT)
```

This command allows the user to work with all locally created alerts in the alert database that are both temporary and permanent. The alerts shown are reporting problems about diskettes.

Error messages for WRKALR

***ESCAPE Messages**

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9812

File &1 in library &2 not found.

CPF9822

Not authorized to file &1 in library &2.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

WRKAPPNSTS (Work with APPN Status) Command Description

WRKAPPNSTS Command syntax diagram

Purpose

Use the Work with APPN Status (WRKAPPNSTS) command to get information about the status of APPN and HPR network sessions, and RTP connections on your local system. The controller (CTL) parameter, the remote network identifier (RMTNETID) parameter, one of either the RMTLOCNAME parameter or the RMTCPNAME parameter, and the TCID parameter when OPTION(*RTP) are used to select the information to display. Note that RMTLOCNAME and RMTCPNAME cannot both be specified.

Optional Parameters

OPTION
Specifies one of the options below.

***SELECT:** A list of options is shown that allows a user to select the information with which to work.

***LOC:** Display the Work with APPN Locations panel.

***RTP:** Display the Work with RTP Connections panel.

CTL Specifies the controller name for which status will be shown. Only sessions using the specified controller are listed on the Work with APPN Status display.

***ALL:** All controllers with active APPN sessions are shown.

generic-controller-name:* Specify the generic name of the controller.

controller-name: Specify the name of the controller.

RMTNETID
Specifies the name of the remote network in which the remote control point or remote location reside.

***ALL:** All remote locations and all remote control points with active APPN sessions are shown. If *ALL is specified for RMTNETID, then RMTCPNAME and RMTLOCNAME must be *ALL.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

remote-network-identifier: Specify the remote network identifier.

RMTLOCNAME
Specifies the remote location name of active APPN sessions for which status is shown. Only sessions with the specified remote location name are listed on the Work with APPN Status display.

***ALL:** For the remote network identifier specified, all remote locations with active APPN sessions are shown.

generic-remote-location-name:* Specify the generic name of the remote location.

remote-location-name: Specify the full name of a remote location.

RMTCPNAME
Specifies the remote control point name of active APPN sessions for which status is shown. Only sessions with the specified remote control point name are listed on the Work with APPN Status display.

For the OPTION(*RTP) view, RMTCPNAME is used to specify the control point name for the RTP connection partner. For the OPTION(*LOC) view, RMTCPNAME is used to specify the control point name for the attached controller.

***ALL:** For the specified network identifier, all remote control points with active APPN sessions are shown.

generic-remote-control-point-name:* Specify the generic name of the remote control point.

remote-control-point-name: Specify the full name of a remote control point.

TCID Specifies the transport connection identifier (TCID) of an RTP connection. Only sessions running over an RTP connection with the specified TCID are listed on the Work with Sessions for RTP Connections panel. This parameter is only valid with OPTION(*RTP).

***ALL:** All TCIDs with active sessions are shown.

transport-connection-identifier: Specify the TCID to show. When the TCID parameter is not equal to *ALL, both the CTL and RMTCPNAME parameters must be *ALL.

MODE Specifies active APPN sessions using the specified mode for which status is shown. Only sessions with the specified mode name are listed on the Work with APPN Status display.

***ALL:** For the specified network identifier, all APPN sessions (regardless of which mode they are using) with active APPN sessions are shown.

***NETATR:** The mode name defined in the network attributes is used.

generic-mode-name:* Specify the generic name of a mode.

mode-name: Specify the name of a mode.

Examples for WRKAPPNSTS

Example 1: Working with RTP Connections

```
WRKAPPNSTS OPTION(*RTP) TCID(*ALL)
```

This command enables the user to display all active RTP connections.

Example 2: Working with APPN Locations

```
WRKAPPNSTS OPTION(*LOC) RMTNETID(ROCV) RMTCPNAME(ROCAS*)
```

For the specified remote control point name, this command allows the user to display all APPN location pairs that have active APPN sessions.

Error messages for WRKAPPNSTS

***ESCAPE Messages** »

WRKASPBRM (Work with ASP Descriptions) 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.

WRKASPBRM Command syntax diagram

Purpose

The Work with ASP Descriptions (WRKASPBRM) command shows the Work with ASP Descriptions display. The display that you see depends on the values that you specify by using the Work with ASP Descriptions command. You can specify an individual ASP, all ASPs, or a generic ASP. You can further define your request by ASP class and the sequence in which you want to sort the resulting output.

Output can be a display or printed report. The report that is produced is the ASP Descriptions report. The report, if printed, is written to printer file QP1AXS.

Example for WRKASPBRM

Example 1: Working with the System ASP

```
WRKASPBRM
ASP(1)
```

In this example you see the Work with ASP Descriptions display. You selected ASP 1, which is the system ASP. From this display you can review or change the description for the selected ASP.

Error messages for WRKASPBRM

No error messages. «

WRKAUT (Work with Authority) Command Description

WRKAUT Command syntax diagram

Purpose

The Work with Authority (WRKAUT) command shows the list of authorized users of an object and their associated authorities. From the list, you can select from options to perform the following:

- Add a user
- Change user authority
- Remove a user

The following are displayed for the specified object:

- The object path name
- The name of the object's owner
- The name of the object's primary group
- A list of all the users who are authorized to use the object
- The authorities that each user has for the object

If an object does not have an owner name associated with it, no authorities for the object are shown.

See Appendix D of the iSeries Security Reference book for the authorities required to use this command.

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.

Required Parameter

OBJ Specifies the path name of the object for which the authorized users and their authorities are to be shown. See path names for more information on specifying path names.

Example for WRKAUT

```
WRKAUT   OBJ('/QSYS.LIB/ARLIB.LIB/PROG1.PGM')
```

This command causes the list of authorized users and their authorities for the object named PROG1 to be shown. PROG1 is a program located in the library named ARLIB.

Error messages for WRKAUT

***ESCAPE Messages**

CPDA080
User profile name too long.

CPE3101
A non-recoverable I/O error occurred.

CPE3408
The address used for an argument was not correct.

CPE3418
Possible APAR condition or hardware failure.

CPE3474
Unknown system state.

CPFA0AA
Error occurred while attempting to obtain space.

CPFA0AB
Object name not a directory.

CPFA0AD
Function not supported by file system.

CPFA0A1
An input or output error occurred.

CPFA0A2
Information passed to this operation was not valid.

CPFA0A3
Path name resolution causes looping.

CPFA0A4
Too many open files for process.

CPFA0A5
Too many open files.

CPFA0A7
Path name too long.

CPFA0A9
Object not found.

CPFA0B1
Requested operation not allowed. Access problem.

CPFA0C0
Buffer overflow occurred.

CPFA0C1
CCSID &1 not valid.

CPFA08B
Path name cannot begin with *.

CPFA08C
Pattern not allowed in path name directory.

CPFA08E
More than one name matches pattern.

CPFA085
Home directory not found for user &1.

CPFA086
Matching quote not found in path name.

CPFA087
Path name contains null character.

CPFA088
Path name pattern not valid.

CPFA09C
Not authorized to object.

CPFA09D
Error occurred in program &1.

CPFA09E
Object in use.

CPFA09F
Object damaged.

CPFA091
Pattern not allowed in user name.

CPFA092
Path name not converted.

CPFA093
Name matching pattern not found.

CPFA094
Path name not specified.

CPF1F05
Directory handle not valid.

CPF1F41
Severe error occurred while addressing parameter list.

CPF1F4A
Value for number of directory entries not valid.

CPF1F53
Value for length of data buffer not valid.

CPF2203
User profile &1 not correct.

CPF2225
Not able to allocate internal system object.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

WRKAUTL (Work with Authorization Lists) Command Description

WRKAUTL Command syntax diagram

Purpose

The Work with Authorization Lists (WRKAUTL) command allows you to work with authorization lists. With this command, you can display, edit, delete, display the list's objects, or change the text for an authorization list.

Restrictions:

1. Only the authorization lists to which you have some authority will be shown on the display.
2. To perform operations on the authorization lists, you must have USE authority to the command used by the operation, and the appropriate authority to the authorization list on which the operation is to be performed.

Required Parameter

AUTL Specifies the name of an authorization list with which you want to work.

***ALL:** A list of all the authorization lists that you either own or have authority to see is shown.

authorization-list-name: Specify the authorization list with which you want to work.

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

Example for WRKAUTL

```
WRKAUTL  AUTL(FR*)
```

This command allows you to work with a list of all the authorization lists that begin with FR that you have authority to see.

Error messages for WRKAUTL

WRKBNDDIR (Work with Binding Directories) Command Description

WRKBNDDIR Command syntax diagram

Purpose

The Work with Binding Directory (WRKBNDDIR) command allows you to display and work with a list of binding directories.

Restrictions:

1. Only the libraries to which you have *USE authority are searched.
2. Only the binding directories to which you have some authority are shown on the display.
3. To perform operations on the binding directories, you must have *USE authority to the command and the appropriate authority to the binding directory on which the operation is performed.

Required Parameter

BNDDIR

Specifies the binding directory to work with.

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

<u>*LIBL:</u> All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

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

***ALL:** Find all binding directories in the specified library or libraries.

binding-directory-name: Specify the name of the binding directory to work with.

generic-directory-name:* Specify the generic name of the binding directories. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all service programs with names that begin with the generic name, and for which the user has authority, are shown. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete service program name.

Example for WRKBNDDIR

```
WRKBNDDIR   BNDDIR(HOLDER)
```

This command allows you to work with a binding directory named HOLDER.

Error Messages for WRKBNDDIR

***Escape Messages**

CPF5D0B
Binding directory &1 was not created.

CPF9809
Library &1 cannot be accessed.

CPF9820
Not authorized to use library &1.

WRKBNDDIRE (Work with Binding Directory Entries) Command Description

WRKBNDDIRE Command syntax diagram

Purpose

The Work with Binding Directory Entries (WRKBNDDIRE) command allows you to work with the entries in a binding directory.

Restrictions:

1. Only the libraries to which you have *USE authority are searched.
2. To perform operations on the binding directories, you must have *USE authority to the command and the appropriate authority to the binding directory on which the operation is to be performed.

Required Parameter

BNDDIR
Work with the entries in the specified binding directory.

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

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

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

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

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

binding-directory-name: Specify the name of the binding directory whose entries are to be shown.

Example for WRKBNDDIRE

```
WRKBNDDIRE   BNDDIR(COINS)
```

This command allows you to work with the entries in binding directory COINS.

Error messages for WRKBNDDIRE

<u>*ESCAPE Messages</u>

CPF5D01
Binding directory &1 in library &2 is not usable.

CPF980F
Binding directory &1 in library &2 not found.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

WRKCALBRM (Work with Calendars 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.

WRKCALBRM Command syntax diagram

Purpose

The Work with Calendars using BRM (WRKCALBRM) command works with all calendar entries by taking you to the Work with Calendars display. From there you can add, remove, change, copy or display calendar information.

Note: Calendar names can be up to 10 characters in length and adhere to OS/400 naming conventions. Output can be a display or printed report. The report that is produced is the Calendars report. The report, if printed, is written to printer file QP1ACA.

Example for WRKCALBRM

Example 1: Working with Backup Control Groups

```
WRKCALBRM
```

In this example you are taken to the Work with Calendars display.

Error messages for WRKCALBRM

None »

WRKCRQD (Work with Change Request Descriptions) Command Description

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

WRKCRQD Command syntax diagram

Purpose

The Work with Change Request Descriptions (WRKCRQD) command shows the Work with Change Request Descriptions display. You can create, change and delete change request descriptions, and work with change request description activities using this command.

Restrictions:

1. Searches only the libraries to which you have *USE authority.
2. Shows only the change request descriptions to which you have *USE authority to.
3. To perform operations on the change request descriptions, you must have *USE authority to the command used by the operation and the appropriate authority to the change request descriptions on which the operation is performed.

Optional Parameter

CRQD Specifies the qualified name of the change request description object.

Element 1: Library Values

***LIBL:** Searches all libraries in your library list and in the system portion of the job's library list.

***CURLIB:** Locate the change request description on the current library. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries listed in the user portion of the library list are searched.

***ALL:** Search all libraries in the system including the QSYS library.

***ALLUSR:** Search all non-system libraries including all user-defined libraries and the QGPL library. Libraries starting with the letter Q other than QGPL are not included.

library-name: Specify the library to be searched in this parameter.

Element 2: Change Request Description Values

***ALL:** Display all change request descriptions.

change-request-description: Specify the change request description to display.

generic-change-request-description:* Specify a generic change request description name. A generic name is a character string containing one or more characters followed by an asterisk (*). Only the change request descriptions with a matching name or with a matching beginning string display.

Examples for WRKCRQD

Example 1: Working with Change Request Descriptions

```
WRKCRQD
```

This command shows how to work with all change request descriptions in the library list of the job.

Example 2: Working with All Change Request Descriptions

```
WRKCRQD CRQD(MYLIB/*ALL)
```

This command shows how to work with all change request descriptions in MYLIB.

Example 3: Working with All Change Request Descriptions in a String

```
WRKCRQD  CRQD(CHG*)
```

This command shows how to work with all change request descriptions that begin with the string CHG and are in the library list of the current job.

Error messages for WRKCRQD

***ESCAPE Messages**

None «

WRKCHTFMT (Work with Chart Formats) Command Description

WRKCHTFMT Command syntax diagram

Purpose

The Work with Chart Formats (WRKCHTFMT) command allows the user to display and work with a list of chart formats from one or more libraries.

Restrictions:

1. Only the libraries to which you have USE authority will be searched.
2. Only the chart formats to which you have some authority will be shown on the display.
3. To perform operations on the chart formats, you must have USE authority to the command used by the operation, and appropriate authority to the chart formats on which the operation is to be performed.

Required Parameter

CHTFMT

Specifies a list of chart formats in the libraries that are shown. If no library qualifier is specified, *LIBL is assumed and all libraries in the job's library list are searched for the chart formats.

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

***LIBL:** All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

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

Notes:

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

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

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

***ALL:** All chart formats in the libraries identified in the library qualifier are shown.

generic-chart-format-name:* Specify the generic name of the chart format. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

chart-format-name: Specify the name of the chart format that is shown.

Example for WRKCHTFMT

```
WRKCHTFMT  CHTFMT(LIB01/ABC*)
```

This command allows you to work with a list of chart formats *beginning* with ABC that are stored in library LIB01.

Error messages for WRKCHTFMT

***ESCAPE Messages**

CPF9809
Library &1 cannot be accessed.

CPF9810>
Library &1 not found.

CPF9820
Not authorized to use library &1.

WRKCOSD (Work with Class-of-Service Descriptions) Command Description

WRKCOSD Command syntax diagram

Purpose

This command shows the Work with Class-of-Service Descriptions display, which allows you to display and work with class-of-service description functions.

Optional Parameter

COSD Specifies the class-of-service descriptions with which you want to work. which class-of-service descriptions to include in the list of class-of-service descriptions on the Work with Class-of-Service Descriptions display.

***ALL:** You can work with all class-of-service descriptions.

generic-COS-name:* Specify the generic name of the COS. 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. See generic names for additional information.

COS-name: Specify a single class-of-service description name.

Example for WRKCOSD

```
WRKCOSD  COSD(MPLS*)
```

This command allows you to utilize the Work with Class-of-Service Descriptions command to display entries for all class-of-service descriptions whose names start with MPLS.

Error messages for WRKCOSD

WRKCLS (Work with Classes) Command Description

WRKCLS Command syntax diagram

Purpose

The Work with Classes (WRKCLS) command allows the user to display and work with a list of classes from one or more libraries.

Restrictions:

1. Only the libraries to which you have USE authority will be searched.
2. Only the classes to which you have some authority will be shown on the display.

3. To perform operations on the classes, you must have USE authority to the command used by the operation, and the appropriate authority to the classes on which the operation is to be performed.

Required Parameter

CLS Specifies a list of class descriptions in the library or libraries shown. If no library qualifier is specified, *LIBL is assumed and all libraries in the job's library list are searched for the class descriptions. More information on this parameter is in commonly used parameters.

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

<u>*LIBL:</u> All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

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

***ALL:** All class descriptions in the libraries identified in the library qualifier are shown (except those class descriptions for which the user does not have some authority).

generic-class-name:* Specify the generic name of the class. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

class-name: Specify the name of the class description that is shown.

Example for WRKCLS

```
WRKCLS  CLS(LIB01/ABC*)
```

This command allows you to display and work with a list of classes beginning with ABC stored in library LIB01.

Error messages for WRKCLS

***ESCAPE Messages**

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

WRKCLSBRM (Work with Classes 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.

WRKCLSBRM Command syntax diagram

Purpose

The Work with Classes using BRM (WRKCLSBRM) command works with all or selected container or media classes. You can select a single class or all classes of media or containers. Output can be a display or printed report.

Example for WRKCLSBRM

Example 1: Working with Container Classes

```
WRKCLSBRM TYPE(*CNR)
```

In this example you are taken to the Work with Container Classes display.

Error messages for WRKCLSBRM

None

WRKCMD (Work with Commands) Command Description

WRKCMD Command syntax diagram

Purpose

The Work with Commands (WRKCMD) command allows you to display and work with a list of commands from a user-specified subset of command names. Several command-related functions are available from this list.

Restrictions:

1. Only the libraries to which you have USE authority will be searched.
2. Only the commands to which you have some authority will be shown on the display.
3. To perform operations on the commands, you must have USE authority to the command used by the operation, and the appropriate authority to the commands on which the operation is to be performed.

Required Parameter

CMD Specifies the qualified name of the command listed on the Work with Commands display. A specific command or a generic command can be specified. Optionally, either type can be qualified by a library name.

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

<u>*LIBL:</u> All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

QDSNX
» QSYS2xxxxx«
QUSROND

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

Notes:

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

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

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

***ALL:** All commands in the specified libraries are listed on the Work with Commands display.

generic-command-name:* Specify the generic name of the command list. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

command-name: Specify the name of the command being listed.

Example for WRKCMD

```
WRKCMD  CMD(QGPL/DSP*)
```

This command allows you to work with a list of all commands in the QGPL library that start with DSP.

Error messages for WRKCMD

***ESCAPE Messages**

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

WRKCMTDFN (Work with Commitment Definitions) Command Description

WRKCMTDFN Command syntax diagram

Purpose

The Work with Commitment Definitions (WRKCMTDFN) command is used to work with the commitment definitions on the system. A commitment definition is used to store information about commitment control when commitment control is started by the Start Commitment Control (STRCMTCTL) command. These commitment definitions may or may not be associated with an active job. Those not associated with an active job have been ended, but one or more of its logical units of work has not yet been completed.

The STATUS parameter can be used to subset the list of commitment definitions by their status. For example, a status value of *RESYNC displays a list of all commitment definitions that are involved with resynchronizing their resources. Commitment definitions may be involved with resychronizing resources in an effort to reestablish a sychronization point across the logical unit of work. A synchronization point is where all resources within a logical unit of work are in consistent state. A status value of *UNDECIDED displays a list of all commitment definitions involved with a commit operation that are waiting to receive the decision to either commit or rollback. A status value of *XOPEN displays a list of all commitment definitions associated with an X/Open global transaction.

» The ASPGRP parameter can be used to subset the list of commitment definition by the auxiliary storage pool (ASP) group on which the commitment definitions reside. «

The logical unit of work identifier (LUWID) parameter can be used when trying to find the commitment definition on the system which is working with a commitment definition on another system. The jobs containing these commitment definitions are communicating using an APPC conversation. An LUWID can be found by displaying the commitment definition on one system and then using it as input to the WRKCMTDFN command to find the corresponding commitment definition.

The duplicate job option (DUPJOBOPT) parameter specifies the action taken when duplicate jobs are found by this command. If duplicate jobs are found they can either be displayed in a list the user can select from or a message can be issued for each duplicate job found.

Optional Parameters

JOB Specifies the names of the jobs (if any) whose commitment definitions are shown. If a job name is not qualified, all jobs by that name have their commitment definitions displayed.

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

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

***:** The commitment definitions which are associated with the job where the WRKCMTDFN command is issued are shown.

***ALL:** Commitment definitions for all jobs on the system are shown.

job-name: Specify the name of the job which is associated with the commitment definitions to be shown.

user-name: Specify the name of the user which is associated with the commitment definitions to be shown.

job-number: Specify the number of the job which is associated with the commitment definitions to be shown.

STATUS

Specifies that only the commitment definitions with a status that matches the value specified on this parameter are listed.

***ALL:** Commitment definitions with all statuses are shown.

***RESYNC:** Only commitment definitions which are involved with resychronizing resources are shown. A commitment definition may be involved with resynchronization in an effort to reestablish a synchronization point. A synchronization point is the point where all resources are in a consistent state.

***UNDECIDED:** Only commitment definitions whose logical unit of work is in a state that is undecided are shown. A commitment definition is in an undecided state when the decision to either commit or rollback resources is unknown to the commitment definition.

» **ASPGRP**

Specifies the name of the auxiliary storage pool (ASP) group on which the commitment definitions to be worked with reside.

***ALLAVL:** Commitment definitions residing on all ASP groups that are currently attached to the system and have a status of ’Available’ are shown.

***SYSBAS:** Commitment definitions residing on the system ASP (ASP number 1) or any configured basic ASP (ASP numbers 2-32) are shown.

auxiliary-storage-pool-group-name: Specify the name of an ASP group that is currently attached to the system and has a status of ’Available’. The ASP group name is the name of the primary ASP device within the ASP group. All commitment definitions that reside on the specified ASP group are shown. «

LUWID

Specifies the logical unit of work identifier of the commitment definition to be shown.

A logical unit of work identifier is a character string made up of three elements:

- Network-qualified logical unit (LU) name
- Instance number
- Sequence number

The network-qualified LU name consists of a character network ID with a maximum of 8 characters, a period delimiter, followed by a LU name with a maximum of 8 characters. The instance number is entered as a 12 character value, each character representing a single hexadecimal digit. The value must be entered in hexadecimal format. For example, X’123456789012’. The sequence number is a decimal value with values ranging from 1 through 65535. For example:

```
APPN.RCHASLGU.X'12578A3BDCFF'.23657
```

***ALL:** Commitment definitions associated with all logical units of work are shown.

generic-logical-unit-of-work-identifier:* Specify the generic name of a logical unit of work identifier. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, all commitment definitions with logical unit of work identifiers that begin with the generic name are shown. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete logical unit of work identifier.

See generic names for additional information.

logical-unit-of-work-identifier: Specify a maximum of 39 characters for the logical unit of work identifier associated with a commitment definition.

OUTPUT

Specifies whether the output from the command is shown at the requesting display station or printed with the job's spooled output.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

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

DUPJOBOPT

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

***SELECT:** The Selection display is shown when duplicate jobs are found during an interactive session. For a batch job, a message is shown.

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

Examples for WRKCMTDFN

Example 1: Limiting by Job

The following example will display a list of commitment definitions associated with the specified job. Of all the commitment definitions on the system, only those associated with the specified job will be listed.

```
WRKCMTDFN   JOB(012345/WULF/WULFS1)
```

Example 2: Limiting by Commitment Definitions Involved with Resynchronizing Resources

The following example will display a list of all of the commitment definitions on the system that are involved with resynchronizing their resources.

```
WRKCMTDFN   JOB(*ALL) STATUS(*RESYNC)
```

Example 3: Limiting by Commitment Definitions That Are Undecided

The following example will display a list of all of the commitment definitions on the system that are in an undecided state. The commitment definitions are in an undecided state when their logical unit of work state is either prepared or last-agent pending.

```
WRKCMTDFN   JOB(*ALL) STATUS(*UNDECIDED)
```

Example 4: Limiting by Commitment Definitions That Are Associated With An X/Open Global Transaction

The following example will display a list of all the commitment definitions associated with an X/Open global transaction.

```
WRKCMTDFN   JOB(*ALL) STATUS(*XOPEN)
```

Example 5: Limiting by LUWID

The following example will display a list of all the commitment definitions whose logical unit of work id begins with the specified generic value. Of all those commitment definitions on the system, only those whose logical unit of work id's begin with the generic value will be listed.

```
WRKCMTDFN   JOB(*ALL) LUWID(APPN.RCHASL7E.X'11223344BDFF'.*)
```

» Example 6: Limiting by system ASP

The following example will display a list of all the commitment definitions that reside on the system ASP.

```
WRKCMTDFN   JOB(*ALL) ASPGRP(*SYSBAS)
```

Example 7: Limiting by ASP device description name

The following example will display a list of all the commitment definitions that reside on an independent ASP.

```
WRKCMTDFN   JOB(*ALL) ASPGRP(IASP0035)
```

«

Error messages for WRKCMTDFN

***ESCAPE Messages**

CPF0941
Job &3/&2/&1 no longer in system.

CPF1069
End of duplicate names.

CPF1070
Job &3/&2/&1 not found.

CPF1071
No authority to job &3/&2/&1.

CPF83E5
Not authorized to jobs.

CPF9845
Error occurred while opening file &1.

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

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

CPF9850
Override of printer file &1 not allowed.

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

CPF9871
Error occurred while processing.

WRKCSI (Work with Communications Side Information) Command Description

WRKCSI Command syntax diagram

Purpose

The Work with Communications Side Information (WRKCSI) command is used to work with side information objects, specified by the side information object name, from the library or libraries specified.

Optional Parameter

CSI Specifies the name of the side information object to work with.

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

***LIBL:** All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

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

Notes:

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

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

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

side-information-name: Specify the name of the object that contains the desired side information. This is the Common Programming Interface (CPI)-Communications symbolic destination name (sym_dest_name). If *LIBL or *USRLIBL is specified as the library name, only the first side information object found with the specified name is listed.

generic-side-information-name:* Specify the generic name of the side information object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

Examples for WRKCSI

Example 1: Displaying Information Objects

```
WRKCSI
```

This command displays all the side information objects in the library list. The user can work with the objects.

Example 2: Displaying Objects the Begin with “SIDE”

```
WRKCSI  CSI(QGPL/SIDE*)
```

This command displays all the side information objects that begin with the characters “SIDE” and are located in library QGPL. The user can work with the objects.

Error messages for WRKCSI

*ESCAPE Messages

WRKCFGL (Work with Configuration Lists) Command Description

WRKCFGL Command syntax diagram

Purpose

The Work with Configuration Lists (WRKCFGL) command allows you to show and work with configuration list functions by using the Work with Configuration Lists display. This command shows the Work with Configuration Lists display.

Optional Parameter

CFGL Specifies the configuration lists with which to work.

***ALL:** Work with all configuration lists.

***APPNDIR:** Work with the advanced peer-to-peer networking (APPN) directory configuration list.

***APPNLCL:** Work with the APPN local location configuration list.

***APPNRMT:** Work with the APPN remote location configuration list.

***APPNSSN:** Work with the APPN session configuration list.

***ASYNCADR:** Work with asynchronous packet assembly/disassembly (PAD) network address configuration lists.

***ASYNCLOC:** Work with the asynchronous remote location configuration list.

***RTLPASTHR:** Work with the retail pass-through configuration list.

***SNAPASTHR:** Work with the SNA pass-through configuration list.

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

list-name: Specify a single configuration list name.

Example for WRKCFGL

```
WRKCFGL  CFGL(PEG*)
```

This command allows you to utilize the Work with Configuration Lists display to work with entries for all configuration lists whose names start with the PEG prefix.

Error messages for WRKCFGL

WRKCFGSTS (Work with Configuration Status) Command Description

WRKCFGSTS Command syntax diagram

Purpose

The Work with Configuration Status (WRKCFGSTS) command is used to display and work with configuration status functions. When this command is run in the interactive mode, the Work with Configuration Status display is shown. This display shows status information for network servers, network interfaces, lines, controllers, and devices, and for jobs associated with devices. The display can be for a location or for one or more network servers, network interfaces, lines, controllers, or devices. All attached configuration descriptions are shown for each configuration object selected. The names of the attached configuration descriptions are indented under the name of the object to which they are attached. If status is requested for a specific configuration object, both upline and downline attachments are shown. If status is requested for any collection of controllers or devices, only downline attachments are shown. Status for a location shows devices and modes for the specified location.

Options available on the Work with Configuration Status display are to vary status, end or resume recovery, hold or release a device, work with a job, or display location path. Options can be entered for more than one configuration object at a time. Option requests are placed in a list and processed sequentially.

For network server descriptions, the status displayed is the configuration status of the object. Options are available from Work with Configuration Status display which shows the status of the network server functions and client sessions. Use the option to check for active clients before varying off a network server.

Required Parameter

CFGTYPE
Specifies the type of description for which to show status.

***NWS:** Status for network servers is shown.

***NWI:** Status for network interfaces is shown.

***LIN:** Status for lines is shown.

***CTL:** Status for controllers is shown.

***DEV:** Status for devices is shown.

Optional Parameters

CFGD Specifies the descriptions being placed on the Work with Configuration Status display.

***ALL:** All descriptions of the type specified for the CFGTYPE parameter are shown.

***APPC:** Status descriptions of advanced program-to-program communications controllers and devices are shown.

***ASYNC:** Status descriptions of asynchronous lines are shown.

***BSC:** Status descriptions of bisynchronous lines are shown.

***CMN:** Status descriptions of communications controllers and/or devices are shown.

***CRP:** Status descriptions of cryptographic devices are shown.

***DDI:** Status descriptions of the distributed data interface devices are shown.

***DSP:** Status descriptions of both local and remote display station devices are shown.

***DKT:** Status descriptions of diskette devices are shown.

***ELAN:** Status descriptions of Ethernet local area network (LAN) lines are shown.

***FAX:** Status descriptions of facsimile (FAX) communication lines are shown.

***FNC:** Status descriptions of finance controllers and devices are shown.

***FR:** Status descriptions of the frame relay lines are shown.

***HOST:** Status descriptions of SNA host controllers and devices are shown.

***IDLC:** Status descriptions of ISDN data link control (IDLC) lines are shown.

***INTRA:** Status descriptions of intrasystem devices are shown.

***ISDN:** Status descriptions of integrated services digital network interfaces are shown.

***LANPRT:** Local area network (LAN) printer devices are displayed.

***LCLDSP:** Status descriptions of local display station devices are shown.

***LCLPRT:** Status descriptions of local printer devices are shown.

***LOC:** Status descriptions of communications devices whose remote location name matches the name specified for the RMTLOCNAME parameter are shown. A value, other than *NONE, must be specified for the RMTLOCNAME parameter if you specify *LOC for the CFGD parameter.

***LWS:** Status descriptions of local work station controllers are shown.

***NET:** Status descriptions of network lines are shown.

***PPP:** Status descriptions of Point-to-Point Protocol (PPP) lines are shown.

***PRT:** Status descriptions of both local and remote printer devices are shown.

***RMTDSP:** Status descriptions of remote display station devices are shown.

***RMTPRT:** Status descriptions of remote printer devices are shown.

***RTL:** Status descriptions of retail controllers and devices are shown.

***RWS:** Status descriptions of remote work station controllers are shown.

***SDLC:** Status descriptions of synchronous data link control (SDLC) lines are shown.

***SNPT:** Status descriptions of SNA pass-through devices are shown.

***TAP:** Status descriptions of tape devices are shown.

***TDLC:** Status descriptions of twinaxial data link control (TDLC) lines are shown.

***TRLAN:** Status descriptions of token ring lines are shown.

***VRTDSP:** Status descriptions of virtual display station pass-through devices are shown.

***VRTPRT:** Status descriptions of virtual (pass-through) printer devices are shown.

***VWS:** Status descriptions of virtual work station pass-through controllers are shown.

***WS:** Status descriptions of work station controllers are shown.

***X25:** Status descriptions of X.25 lines are shown.

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

description-name: Specify display of a specific description and any upline attachments as well as downline attachments.

ASTLVL

Specifies the user interface to use.

***PRV:** The previous user interface is used.

***USRPRF:** The user interface specified in the user profile is used.

***BASIC:** The Operational Assistant* user interface is used.

***INTERMED:** The system interface is used.

***ADVANCED:** The expert system interface is used.

RANGE

Specifies whether downline or upline attached configuration descriptions are shown.

***NET:** If a name for a single description is specified for the configuration description prompt (CFGD parameter), both downline and upline descriptions are shown. If a special value or generic name is specified for the CFGD parameter, downline descriptions are shown.

***OBJ:** Only objects of the type specified by the configuration type prompt (CFGTYPE parameter) are shown.

OUTPUT

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

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

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

RMTLOCNAME

Specifies the remote location name of communications devices for which status is shown. Only devices with the specified remote location name are listed on the Work with Configuration Status display.

This parameter is required if CFGD(*LOC) is specified. It is not a valid parameter for any other value of CFGD.

***NONE:** The descriptions shown are not for communications devices with a specific location name. *NONE must be specified if a value other than *LOC is specified on the CFGD parameter.

generic-remote-location-name:* Specify the generic name of the remote location.

remote-location-name: Specify the full name of a remote location.

STATUS

Specifies the status values used to subset the list descriptions shown.

***ALL:** All descriptions are included in the list regardless of their status.

***ACTIVE:** All descriptions with an active status are shown.

***FAILED:** All descriptions with a failed, recovery, damaged, or unknown status are shown.

***VARYOFF:** All descriptions with a varied off status are shown.

***VARYON:** All descriptions with a varied on status are shown.

Examples for WRKCFGSTS

Example 1: Showing the Status for All Remote Display Stations

```
WRKCFGSTS  CFGTYPE(*DEV)  CFGD(*RMTDSP)
```

This command uses the Work with Configuration Status display to show the status for all remote display stations.

Example 2: Showing the Status for All Network Servers

```
WRKCFGSTS  CFGTYPE(*NWS)  CFGD(*ALL)
```

This command allows the user to utilize the Work with Configuration Status command to show the status for all network servers on the system.

Error messages for WRKCFGSTS

*ESCAPE Messages

CPF1E99

Unexpected error occurred.

CPF2602

Controller &1 not found.

CPF2702

Device description &1 not found.

CPF2703

Controller description &1 not found.

CPF2704

Line description &1 not found.

CPF9846

Error while processing file &1 in library &2.

WRKCNNLE (Work with Connection List Entries) Command Description

WRKCNNLE Command syntax diagram

Purpose

The Work with Connection List Entries (WRKCNNLE) command allows the user to work with specific connection list entries.

Required Parameter

CNNL Specify the connection list containing the entries to work with.

Example for WRKCNNLE

```
WRKCNNLE  CNNL(CHICAGO)
```

This command allows the user to use the Work with Connection List Entries display to work with the entries in connection list CHICAGO.

Error messages for WRKCNNLE

***ESCAPE Messages**

CPF2625
Not able to allocate object &1.

CPF2634
Not authorized to object &1.

CPF266C
Connection list &1 not found.

CPF266D
Program name &1 not found in system library.

CPF266E
Connection list &1 has been damaged.

WRKCNNL (Work with Connection Lists) Command Description

WRKCNNL Command syntax diagram

Purpose

The Work with Connection Lists (WRKCNNL) command allows the user to work with a connection list.

Optional Parameter

CNNL Specifies the connection list to work with.

***ALL:** All connection lists are worked with.

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

connection-list-name: Specify a specific connection list name.

Example for WRKCNNL

```
WRKCNNL  CNNL(CHI*)
```

This command allows the user to use the Work with Connection List display to work with connection lists which have names that begin with ’CHI’ and for which the user has authority.

Error messages for WRKCNNL

***ESCAPE Messages**

CPF2625
Not able to allocate object &1.

CPF2634
Not authorized to object &1.

CPF266C
Connection list &1 not found.

CPF266D
Program name &1 not found in system library.

CPF266E
Connection list &1 has been damaged.

WRKCNTINF (Work with Contact Information) Command Description

WRKCNTINF Command syntax diagram

Purpose

The Work with Contact Information (WRKCNTINF) command allows you to display and work with the information that helps you contact, or be contacted by, various support centers. When this command is specified, a display appears that allows selection of one of six kinds of support functions.

More information on this command is in the Use tape library article in the Information Center.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QSRV and QSRVBAS user profiles have private authorities to use the command.
2. This command is valid only in an interactive environment.

There are no parameters for this command.

Example for WRKCNTINF

```
WRKCNTINF
```

This command allows you to work with the Work with Support Contact Information display.

Error messages for WRKCNTINF

***ESCAPE Messages**

CPF8C84
Error detected while processing support contact data.

CPF8C96
Description is a required field.

CPF8C97
Description already exists in system directory.

WRKCNRBRM (Work with Containers 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.

WRKCNRBRM Command syntax diagram

Purpose

The Work with Containers using BRM (WRKCNRBRM) command works with all or selected containers. You can select all containers or those that are open or closed. Your selection can be further refined by location. Output can be a display or printed report. The report that is produced is the Container report. The report, if printed, is written to printer file QP1ACN.

Example for WRKCNRBRM

Example 1: Working with Containers

```
WRKCNRBRM STATUS(*OPEN) LOC(*HOME)
```

In this example you are taken to the Work with Containers display where all open containers that are at the home location are displayed.

Error messages for WRKCNRBRM

None

WRKCTLGBRM (Work with Control Groups 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.

WRKCTLGBRM Command syntax diagram

Purpose

The Work with Control Groups using BRM (WRKCTLGBRM) command works with backup and archive control groups. You are taken to the selected Work with Control Groups display to create, delete, change or display control groups.

Example for WRKCTLGBRM

Example 1: Working with Backup Control Groups

```
WRKCTLGBRM
```

In this example you are taken to the Work with Backup Control Groups display.

Error messages for WRKCTLGBRM

None

WRKCTLD (Work with Controller Descriptions) Command Description

WRKCTLD Command syntax diagram

Purpose

The Work with Controller Descriptions (WRKCTLD) command allows you to display and work with controller description functions by using the Work with Controller Descriptions display. This command shows the Work with Controller Descriptions display, which displays a list of controller descriptions to which you are authorized.

From the Work with Controller Descriptions display, you can use all of the available configuration functions with your controller descriptions. You can create, change, copy, delete, display, and print your controller descriptions.

This command also allows you to group the controller descriptions that appear on the Work with Controller Descriptions display. You can have all of the controller descriptions, only those controller descriptions of a certain type, or only those controller descriptions whose names start with a certain character string on the display.

Optional Parameter

CTLD Specifies which controller descriptions to include in the list of controller descriptions on the Work with Controller Descriptions display.

***ALL:** The user works with all controller descriptions.

***CMN:** The user works with communications controller descriptions only.

***WS:** The user works with work station controller descriptions only.

***TAP:** The user works with tape controller descriptions only.

***LWS:** The user works with local work station controller descriptions only.

***RWS:** The user works with remote work station controller descriptions only.

***VWS:** The user works with virtual work station controller descriptions only.

generic-controller-name:* Specify the generic name of the controller. 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. See generic names for additional information.

controller-name: Specify the name of a controller description.

Example for WRKCTLD

```
WRKCTLD  CTLD(*LWS)
```

This command allows you to use the Work with Controller Descriptions display to work with entries for all local work station controllers to which you have authority.

Error messages for WRKCTLD

***ESCAPE Messages**

WRKDTAARA (Work with Data Areas) Command Description

WRKDTAARA Command syntax diagram

Purpose

The Work with Data Areas (WRKDTAARA) command allows the user to display and work with a list of data areas from one or more libraries.

Restrictions:

1. Only the libraries to which you have USE authority will be searched.
2. Only the data areas to which you have some authority will be shown on the display.
3. To perform operations on the data areas, you must have USE authority to the command used by the operation, and the appropriate authority to the data areas on which the operation is to be performed.

Required Parameter

DTAARA

Specifies a list of data areas in the libraries that are shown. If no library qualifier is specified, *LIBL is assumed and all libraries in the job's library list are searched for the data areas.

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

<u>*LIBL:</u> All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

» QRCLxxxxx«
» QSYS2«
QUSRINFSKR
QUSRNOTES

Notes:

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

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

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

***ALL:** All data areas in the libraries identified in the library qualifier are shown.

generic-data-area-name:* Specify the generic name of the data area. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

data-area-name: Specify the name of the data area that is shown.

Example for WRKDTAARA

```
WRKDTAARA   DTAARA(LIB01/ABC*)
```

This command allows you to display and work with a list of data areas beginning with ABC stored in library LIB01.

Error messages for WRKDTAARA

<u>*ESCAPE Messages</u>

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

WRKDTADFN (Work with Data Definitions) Command Description

WRKDTADFN Command syntax diagram

Purpose

The Work with Data Definitions (WRKDTADFN) command allows you to create, change, copy, delete, print, or rename definitions, or display where definitions are used. If the definition type is not specified, the Interactive Data Definition Utility (IDDU) Select Definition Type display is shown. The dictionary and type of definition to process is selected from this display.

Optional Parameters

DTADCT

Specifies the data dictionary being used.

***PRV:** The name used in the previous session is used.

data-dictionary-name: Specify the data dictionary name to use.

DFNTYPE

Specifies the data definition type being used.

***ALL:** Users select the definition type and data dictionary from a list of all the data dictionaries and definition types.

***FILE:** Users work with the definition type from a list of all file definitions in the data dictionary specified.

***RCDFMT:** Users work with record format definitions in the data dictionary specified.

***FLD:** Users work with field definitions in the data dictionary specified.

Example for WRKDTADFN

```
WRKDTADFN   DFNTYPE(*FILE)
```

This command allows you to work with file definitions in the data dictionary you worked with last.

Error messages for WRKDTADFN

***ESCAPE Messages**

WRKDTADCT (Work with Data Dictionaries) Command Description

WRKDTADCT Command syntax diagram

Purpose

The Work with Data Dictionaries (WRKDTADCT) command allows you to work with the Interactive Data Definition Utility (IDDU) Work with Data Dictionaries display on which you can select an option to create, change, delete, or print the contents of a data dictionary.

There are no parameters for this command.

Example for WRKDTADCT

```
WRKDTADCT
```

This command allows you to work with the Work with Data Dictionaries display.

Error messages for WRKDTADCT

***ESCAPE Messages**

WRKDTAQ (Work with Data Queues) Command Description

WRKDTAQ Command syntax diagram

Purpose

The Work with Data Queues (WRKDTAQ) command allows the user to display and work with a list of data queues from one or more libraries.

Restrictions:

1. Only the libraries to which you have USE authority will be searched.
2. Only the data queues to which you have some authority will be shown on the display.
3. To perform operations on the data queues, you must have USE authority to the command used by the operation, and the appropriate authority to the data queues on which the operation is to be performed.

Required Parameter

DTAQ Specifies a list of data queues in the libraries that are shown. If no library qualifier is specified, *LIBL is assumed and all libraries in the job's library list are searched for the data queues.

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

<u>*LIBL:</u> All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

QSYS2	QUSRNOTES

Notes:

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

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

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

***ALL:** All data queues in the libraries identified in the library qualifier are shown.

generic-data-queue-name:* Specify the generic name of the data queue. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be worked with only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

data-queue-name: Specify the name of the data queue that is shown.

Example for WRKDTAQ

```
WRKDTAQ  DTAQ(LIB01/ABC*)
```

This command allows you to display and work with a list of data queues beginning with ABC stored in library LIB01.

Error messages for WRKDTAQ

***ESCAPE Messages**

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

WRKDBFIDD (Work with Database Files Using IDDU) Command Description

WRKDBFIDD Command syntax diagram

Purpose

The Work with Database Files Using the Interactive Data Definition Utility (IDDU) (WRKDBFIDD) command allows you to select an option from the IDDU Work with Database Files display to create physical files or enter data into a file.

Optional Parameter

LIB Specifies the name of the library that contains the files.

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

***PRV:** The files are located in the last library the user worked with in the IDDU. If this is the first time the user works with the IDDU, the current library is used.

***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.

Example for WRKDBFIDD

```
WRKDBFIDD  DEPT245
```

This command allows you to work with database files in the DEPT245 library using the IDDU Work with Database Files display.

Error messages for WRKDBFIDD

***ESCAPE Messages**

WRKDEVD (Work with Device Descriptions) Command Description

WRKDEVD Command syntax diagram

Purpose

The Work with Device Descriptions (WRKDEVD) command displays and work with device description functions. When this command is entered, the Work with Device Descriptions display is shown. This display consists of a list of device descriptions to which you are authorized.

From the Work with Device Descriptions display, you can perform all of the available configuration functions on the device descriptions. You can create, change, copy, delete, display, and print the device descriptions.

This command also allows you to group the device descriptions that appear on the Work with Device Descriptions display. Group displays can include all of the device descriptions of only a certain type of device, or only descriptions of devices whose names start with a certain character string prefix.

Optional Parameters

DEVD Specifies the name of the device descriptions being listed on the Work with Device Descriptions display.

***ALL:** All device descriptions are shown.

***CMN:** Communications devices are shown.

***CRP:** Cryptographic devices are shown.

***DKT:** Diskette devices are shown.

***DSP:** Both local and remote display station devices are shown.

***LCLDSP:** Local display station devices are shown.

***RMTDSP:** Remote display station devices are shown.

***VRTDSP:** Virtual display station pass-through devices are shown.

***LOC:** Devices whose remote location name matches the name specified on the RMTLOCNAME parameter are shown. A value, other than *NONE, must be specified for the RMTLOCNAME parameter if you specify *LOC for the DEVD parameter.

***PRT:** Both local and remote printer devices are shown.

***LANPRT:** Printer devices attached to a local area network (LAN) are shown.

***LCLPRT:** Local printer devices are shown.

***RMTPRT:** Remote printer devices are shown.

***VRTPRT:** Virtual (passthrough) printer devices are shown.

***TAP:** Tape devices are shown.

***SNPT:** SNA pass-through devices are shown.

generic-device-name:* Specify the generic name of the device. 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. See generic names for additional information.

device-name: Specify name of the device descriptions to be shown.

RMTLOCNAME

Specifies the remote location name for communications devices being shown. Only device descriptions with the specified remote location name are listed on the Work with Device Descriptions display.

This parameter is required if DEVD(*LOC) is specified and is not valid if DEVD(*LOC) is not specified.

***NONE:** The descriptions shown are not for a specific location. *NONE must be specified if the DEVD parameter is other than *LOC.

generic-remote-location-name:* Specify the generic name of the remote location.

remote-location-name: Specify the remote location name of the communications devices to be shown.

Example for WRKDEVD

```
WRKDEVD  DEVD(*LCLPRT)
```

This command uses the Work with Device Descriptions display of all the local printers to which you have authority.

Error messages for WRKDEVD

***ESCAPE Messages**

Printed in U.S.A.