System i

Programming
i5/OS commands
Starting with ENDCLNUP (End Cleanup)

Version 6 Release 1

System i

Programming
i5/OS commands
Starting with ENDCLNUP (End Cleanup)

Version 6 Release 1

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 405.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

Contents

Install Program Temporary Fix (INSPTF) 345

Install Windows Server (INSWNTSVR) 351

Initialize DLFM (INZDLFM) 389

Initialize Distribution Queue (INZDSTQ). 391

Initialize NWS Configuration (INZNWSCFG) 395

Initialize Optical (INZOPT) 399

Appendix. Notices 405

End Cleanup (ENDCLNUP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Cleanup (ENDCLNUP) command allows you to end the cleanup operation. The cleanup operation allows items on the system to be deleted automatically after they are a specified number of days old. Any active batch cleanup jobs, either processing or on the job queue, are ended immediately.

This command does not alter any of the parameters specified on the Change Cleanup (CHGCLNUP) command. The cleanup operation can be restarted by specifying the Start Cleanup (STRCLNUP) command.

More information is in the Basic system operations topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restriction: You must have job control (*JOBCTL) special authority to use this command.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDCLNUP
```

This command ends the cleanup operation.

Top

Error messages

***ESCAPE Messages**

CPF1E2A
Unexpected error in QSYSSCD job.

CPF1E2B
Power scheduler and cleanup options not found.

CPF1E33
Cleanup options or power schedule in use by another user.

CPF1E35
Not authorized to end cleanup.

CPF1E36
Cleanup has not been started.

CPF1E99
Unexpected error occurred.

Top

End Communications Server (ENDCMNSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Communications Server (ENDCMNSVR) command is used to end the target display station pass-through server or the display station pass-through utilities server. The target display station pass-through server processes display station pass-through, System i5 Access work station function (WSF), and other 5250 emulation programs on programmable workstations. The display station pass-through utilities server can reduce the time required to establish a session.

Restriction: You must have job control (*JOBCTL) special authority to use this command.

Top

Parameters

Keyword	**Description**	**Choices**	**Notes**
SERVER	Server type	***PASTHR**, *UTILSVR	Optional, Positional 1
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 2
DELAY	Controlled end delay time	1-86400, ***NOMAX**	Optional

Top

Server type (SERVER)

Specifies the type of server to be ended.

***PASTHR**
The target display station pass-through server, if active, is ended.

***UTILSVR**
The display station pass-through utilities server, if active, is ended.

Top

How to end (OPTION)

Specifies whether the target display station pass-through server or the display station pass-through utilities server is ended in an immediate or controlled manner.

***CNTRLD**
The server is ended in a controlled manner. Active sessions are allowed to complete their processing. New sessions are not allowed. After the specified period of time elapses, the processing for ENDCMNSVR OPTION(*IMMED) is performed.

***IMMED**
The server is ended in an immediate fashion. All active sessions that were started through the target display station pass-through server are ended immediately.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed in which to complete a controlled end of the target display station pass-through server. After this period of time all the target display station pass-through server jobs are ended immediately.

***NOMAX**
There is no maximum amount of time to wait. The servers will not end until all active sessions end normally.

delay-time
Specify the number of seconds in which the end operation is completed. Valid values range from 1 through 86400 seconds.

Top

Examples

Example 1: Ending Target Display Station Pass-through Server

```
ENDCMNSVR
```

This command ends the target display station pass-through server in a controlled manner. Any active sessions that are using the target display station pass-through server are not affected. New sessions are not allowed through the target display station pass-through server. Once all of the active sessions have ended, the target display station pass-through server will end.

Example 2: Ending Display Station Pass-through Utilities Server

```
ENDCMNSVR SERVER(*UTILSVR)
```

This command ends the display station pass-through utilities server in a controlled manner. Any active sessions that are using the display station pass-through utilities server are not affected. The time required to establish new sessions may increase.

Top

Error messages

***ESCAPE Messages**

CPF8947
Unexpected error ending target display station pass-through servers.

Top

End Communications Trace (ENDCMNTRC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Communications Trace (ENDCMNTRC) command ends the trace running on the specified line, network interface, or network server description.

Restrictions:

- You must have use (*USE) authority to the line, network interface or network server to be traced.
- You must have service (*SERVICE) special authority, or be authorized to the Service trace function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, 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.
- The following user profiles have authority to this command:
  - QSECOFR
  - QSRV

Top

Parameters

Keyword	Description	Choices	Notes
CFGOBJ	Configuration object	*Name*	Required, Positional 1
CFGTYPE	Type	*LIN, *NWI, *NWS	Required, Positional 2

Top

Configuration object (CFGOBJ)

Specifies the configuration object being traced. The object is either a line description, or a network interface description, or a network server description.

This is a required parameter.

name Specify the name of the configuration object for which communications tracing is to be ended.

Top

Type (CFGTYPE)

Specifies the type of configuration description being traced.

This is a required parameter.

***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.

Top

Examples

```
ENDCMNTRC   CFGOBJ(*QESLINE)  CFGTYPE(*LIN)
```

This command ends the communications trace of line description QESLINE.

Top

Error messages

***ESCAPE Messages**

CPF2601
Line description &1 not found.

CPF2634
Not authorized to object &1.

CPF26AE
Network server description &1 not found.

CPF39AE
Trace already ended.

CPF39AF
Trace is ending - please wait

CPF39A7
Trace storage not available in communications processor

CPF39A8
Not authorized to communications trace service tool

CPF39A9
Error occurred during communications trace function

CPF39BD
Network interface description &1 not found

CPF39B0
No communications traces exist.

CPF39B1
Trace &1 type &2 does not exist

CPF39B6
Communications trace function cannot be performed

CPF39C3
Trace &1 type &2 cannot be ended.

CPF98A2
Not authorized to &1 command or API.

Top

End Commitment Control (ENDCMTCTL)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The End Commitment Control (ENDCMTCTL) command ends the commitment definition associated with the activation group for the program that issued the command. Changes to commitment resources associated with the commitment definition are no longer made after this command is processed.

This command either ends the activation group level or the job level commitment definition associated with the activation group for the program that issued the command. A commitment definition is first established by the Start Commitment Control (STRCMTCTL) command.

If there are uncommitted changes for an interactive job, a message is sent asking the user whether the changes should be committed or rolled back before a commitment definition is ended. For a batch job, the changes are rolled back.

More information on the ENDCMTCTL command is in the Commitment control topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDCMTCTL
```

This command specifies that the commitment definition established with the STRCMTCTL command is to end. The system determines if any changes have been made to the commitment resources after the last commitment boundary (at the last completed Commit (COMMIT) command or Rollback (ROLLBACK) command). If changes have been made for an interactive job, a message is sent asking the user whether the changes should be made permanent (committed) or removed (rolled back). For batch jobs, any changes are rolled back.

Top

Error messages

***ESCAPE Messages**

CPF83E4
Commitment control ended with resources not committed.

CPF835A
End of commitment definition &1 canceled.

CPF835B
Errors occurred while ending commitment control.

CPF835C
Commitment control ended with remote changes not committed.

CPF8350
Commitment definition not found.

CPF8355
ENDCMTCTL not allowed. Pending changes active.

CPF8356
Commitment control ended with &1 local changes not committed.

CPF8367
Cannot perform commitment control operation.

Top

End Copy Screen (ENDCPYSCN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Copy Screen (ENDCPYSCN) command ends the copy screen image operation for the specified display device.

Note: The target display station can also stop the copy screen image operation if the user presses the System Request key and types ENDCPYSCN on the command line. No parameters can be specified.

Top

Parameters

Keyword	Description	Choices	Notes
SRCDEV	Source device	*Name*, ***REQUESTER**	Optional, Positional 1

Top

Source device (SRCDEV)

Specifies the display device that is currently having its screen images copied.

***REQUESTER**
Ends the copy screen image operation for the display device running this command.

name Specifies the device name of the display station that is having its screen images copied.

Top

Examples

```
ENDCPYSCN   SRCDEV(CHARLIE)
```

The command sends a message to ’CHARLIE’ (the source display station). The message indicates the copy screen image operation is about to end. The target work station display is restored to the same display image that was shown before the operation started. The sign-on display is normally shown.

Top

Error messages

***ESCAPE Messages**

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF7AF7
Device name &1 not correct.

CPF7AF8
Device name &1 not being copied.

Top

End Controller Recovery (ENDCTLRCY)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Controller Recovery (ENDCTLRCY) command ends automatic error recovery procedures for a specific controller. If any type of failure occurs after this command is run, an inquiry message is sent to the system operator.

Use the Resume Controller Recovery (RSMCTLRCY) command to reestablish error recovery procedures for the controller.

Top

Parameters

Keyword	Description	Choices	Notes
CTL	Controller	*Name*	Required, Positional 1

Top

Controller (CTL)

Specifies the controller whose recovery is to be ended.

This is a required parameter.

Top

Examples

```
ENDCTLRCY   CTL(TROLL3)
```

This command ends error recovery procedures for the controller TROLL3.

Top

Error messages

***ESCAPE Messages**

CPF2703
Controller description &1 not found.

CPF5924
Controller &1 does not allow automatic error recovery.

CPF5928
Controller &1 not varied on.

CPF5929
Controller &1 assigned to another job.

CPF5935
Error occurred during command processing.

CPF5936
Not authorized to controller &1.

Top

End Debug Mode (ENDDBG)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Debug (ENDDBG) command ends debug mode for a job, removes all breakpoints and traces, clears any trace data, and removes all programs from debug mode. This command cannot be entered when one or more of the programs in the call stack are stopped at a breakpoint. All breakpoints must be canceled by Resume Breakpoint (RSMBKP) or End Request (ENDRQS) commands. After this command has been entered, all database files in production libraries can be updated normally.

If ENDDBG is not done before the job has ended, all trace data is printed.

Restriction: This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.

If you are servicing another job and you are operating in debug mode, this command must be specified before you can use the End Service Job (ENDSRVJOB) command.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDDBG
```

Assuming that this command is entered interactively and no program in the call stack is stopped at a breakpoint, debug mode for the job is ended.

Top

Error messages

***ESCAPE Messages**

CPF1931
Command not valid at this time.

CPF1999
Errors occurred on command.

Top

End Debug Server (ENDDBGSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Debug Server (ENDDBGSVR) command ends the debug server router function. If there are active server jobs running when the router function is ended, the servers remain active until the connection with the client is ended. Subsequent connection requests fail until the debug server router function is started again.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDDBGSVR
```

This command ends the debug server router function.

Top

Error messages

None

Top

End Database Monitor (ENDDBMON)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End Database Monitor (ENDDBMON) command ends the collection of database performance statistics for a specified job, all jobs on the system or a selected set of jobs (i.e. a generic job name).

To end a monitor, you can specify the job or the monitor ID or both. If just the JOB parameter is specified, we end the monitor that was started using the same exact JOB parameter - if there is only one monitor which matches the specified JOB. If more than one monitor is active which matches the specified JOB, then the user uniquely identifies which monitor is to be ended by use of the MONID parameter. When just the MONID parameter is specified, we compare the specified MONID to the monitor ID of the monitor for the current job and to the monitor ID of all active public monitors (monitors that are open across multiple jobs). The monitor matching the specified MONID is closed.

The monitor ID is returned in the informational message CPI436A which is generated for each occurrence of the STRDBMON command. The monitor ID can also be found in column QQC101 of the QQQ3018 database monitor record.

Restrictions:

- You cannot end database monitoring for a specific job by using JOB(*ALL) on the ENDDBMON command. To end a specific job you must specify that job on the JOB parameter or specify JOB(*).
- If JOB(*ALL) was specified on the Start Database Monitor (STRDBMON) command, you cannot end monitoring on a per-job basis. Specifying ENDBMON JOB(*ALL) will end the monitor that is active across all jobs. Only if a specific job was specified on the JOB parameter or JOB(*) can you end monitoring for a specific job.
- This command is conditionally threadsafe. It is not threadsafe (and may fail) when the OUTFILE parameter for the STRDBMON command specified a distributed file or a Distributed Data Management (DDM) file of type *SNA.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	Single values: <u>*</u> Other values: *Qualified job name*	Optional, Positional 1
	Qualifier 1: Job name	*Generic name, name*, *ALL	
	Qualifier 2: User	*Generic name, name*, *ALL	
	Qualifier 3: Number	000000-999999, *ALL	
MONID	Monitor ID	*Character value*, *ALL	Optional
COMMENT	Comment	*Character value*, <u>***BLANK**</u>	Optional

Top

Job name (JOB)

Specifies the job(s) for which the database monitor is to be ended.

Single values

***** The data monitor for the job running the ENDDBMON command is to be ended.

***ALL** The data monitor open across all jobs (started with JOB(*ALL)) is to be ended.

Qualifier 1: Job name

name Specify the name of the job whose database monitor is to be ended. If no job user name or job number qualifiers are specified, all of the jobs currently in the system are searched for the specified simple job name. If duplicates of the specified job name are found, you need to specify a job user name, job number or monitor ID that uniquely identifies the job to be changed.

generic-name
Specify the generic job name of the monitor to be ended. The monitor started with the same generic job name (e.g. JOB(QZDA*)) will be ended.

Qualifier 2: User

name Specify the name of the user of the job whose database monitor is to be ended.

generic-name
Specify the generic job user of the monitor to be ended. The monitor started with the same generic job user (e.g. JOB(*ALL/DEVLPR*)) will be ended.

Qualifier 3: Number

000000-999999
Specify the number of the job whose database monitor is to be ended.

Top

Monitor ID (MONID)

Specifies the unique monitor ID of the database monitor to be ended. If mulitple monitors using the same JOB parameter were started, the MONID parameter must be specified to uniquely identify which individual monitor is to be ended. A monitor ID value is generated by the system for each invocation of the STRDBMON command. Informational message CPI436A contains the system generated monitor ID value. The system generated monitor ID value is also stored in column QQC101 of the QQQ3018 database monitor record.

***ALL** When used with JOB(*ALL), all public monitors (monitors over multiple jobs) will be ended. When used with a generic job name (e.g. JOB(QZDA*)), all public monitors (monitors over multiple jobs) started with JOB(QZDA*) will be ended.

simple-name
Specify the 10-byte monitor ID of the monitor that is to be ended.

Top

Comment (COMMENT)

User-specified description that is associated with the database monitor. The description is stored in the monitor record that has a record ID of 3018.

***BLANK**
Text is not specified.

character-value
Specify up to 100 characters of text.

Top

Examples

Example 1: End Monitoring for a Specific Job

```
ENDDBMON   JOB(*)
```

This command ends database monitoring for the current job.

Example 2: End Monitoring for All Jobs

```
ENDDBMON   JOB(*ALL)
```

This command ends the monitor open across all jobs on the system. If more than one monitor with JOB(*ALL) is active, then the MONID parameter must also be specified to uniquely identify which individual monitor to end.

Example 3: End Monitoring for an Individual Public Monitor with MONID Parameter

```
ENDDBMON   JOB(*ALL) MONID(061601001)
```

This command ends the monitor that was started with JOB(*ALL) and that has a monitor ID of 061601001. Because there were mulitple monitors started with JOB(*ALL), the monitor ID must be specified to uniquely identify which monitor started with JOB(*ALL) is to be ended.

Example 4: End Monitoring for an Individual Public Monitor with MONID Parameter

```
ENDDBMON   MONID(061601001)
```

The commands performs the same function as the previous example. It ends the monitor that was started with JOB(*ALL) or JOB(*) and that has a monitor ID of 061601001.

Example 5: End Monitoring for All JOB(*ALL) Monitors

```
ENDDBMON   JOB(*ALL/*ALL/*ALL) MONID(*ALL)
```

This command ends all monitors that are active across multiple jobs. It will not end any monitors open for a specific job or the current job.

Example 6: End Monitoring for a Generic Job

```
ENDDBMON   JOB(QZDA*)
```

This command ends the monitor that was started with JOB(QZDA*). If more than one monitor with JOB(QZDA*) is active, then the MONID parameter must also be specified to uniquely identify which individual monitor to end.

Example 7: End Monitoring for an Individual Monitor with a Generic Job

```
ENDDBMON   JOB(QZDA*) MONID(061601001)
```

This command ends the monitor that was started with JOB(QZDA*) and has a monitor ID of 061601001. Because there were mulitple monitors started with JOB(QZDA*), the monitor ID must be specified to uniquely identify which JOB(QZDA*) monitor is to be ended.

Example 8: End Monitoring for a Group of Generic Jobs

```
ENDDBMON   JOB(QZDA*) MONID(*ALL)
```

This command ends all monitors that were started with JOB(QZDA*).

Top

Error messages

***ESCAPE Messages**

CPF1321
Job &1 user &2 job number &3 not found.

CPF436D
Job &1 is not being monitored.

CPF436E
Job &1 user &2 job number &3 is not active.

CPF43A1
Job &1 specified on ENDDBMON command is not unique.

Top

End Device Recovery (ENDDEVRCY)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Device Recovery (ENDDEVRCY) command ends automatic error recovery procedures for a specific device. If any type of failure occurs after this command is run, an inquiry message is sent to the system operator. The user must have object operational authority for the device.

Use the Resume Device Recovery (RSMDEVRCY) command to reestablish error recovery procedures for the device.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	*Name*	Required, Positional 1

Top

Device (DEV)

Specifies the device whose recovery is to be ended. Specify the name specified for the device in the device description.

Top

Examples

```
ENDDEVRCY   DEV(WSPR03)
```

This command ends error recovery procedures for the device WSPR03.

Top

Error messages

***ESCAPE Messages**

CPF5923
Device &1 does not allow automatic error recovery.

CPF5925
Device &1 not varied on.

CPF5935
Error occurred during command processing.

CPF9814

Device &1 not found.

Top

End Directory Shadowing (ENDDIRSHD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Directory Shadowing (ENDDIRSHD) command ends the directory shadow controlling job in the system work subsystem (QSYSWRK).

Any active collector or supplier jobs running are allowed to complete. No new collector jobs are started. Supplier jobs are prevented from starting if a collector system requests data through directory shadowing. The Start Directory Shadowing (STRDIRSHD) command can be used to re-start directory shadowing.

Restriction: You must have job control (*JOBCTL) authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 1
DELAY	Controlled end delay time	1-999999, **30**	Optional, Positional 2

Top

How to end (OPTION)

Specifies whether the directory shadow controlling job is ended in a controlled manner or immediately.

***CNTRLD**
The directory shadow controlling job is ended in a controlled manner. This allows the directory shadow controlling job to perform cleanup (end-of-job processing).

***IMMED**
The directory shadow controlling job is ended immediately. The directory shadow controlling job is not allowed to perform any cleanup.

Note: Using the *IMMED option can cause unexpected results if data has been only partially updated.

This is a required parameter.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed for the directory shadow controlling job to complete its cleanup processing during a controlled end. This parameter is not valid if OPTION(*IMMED) is specified. If the cleanup is not complete before the end of the delay time, the directory shadow controlling job is immediately ended.

30
A maximum delay time of 30 seconds is allowed for cleanup before the directory shadow controlling job is ended.

delay-time
Specify the maximum amount of delay time in seconds before the directory shadow controlling job is ended. Valid values range from 1 through 999999.

This is a required parameter.

Top

Examples

Example 1: Ending Directory Shadowing in a Controlled Manner

```
ENDDIRSHD   OPTION(*CNTRLD)  DELAY(60)
```

The directory shadow controlling job is ended in the system work subsystem in a controlled manner and will have 60 seconds to complete its end-of-job processing.

Example 2: Ending Directory Shadowing Immediately

```
ENDDIRSHD   OPTION(*IMMED)
```

The directory shadow controlling job is ended in the system work subsystem immediately. The directory shadow controlling job does not perform end-of-job processing.

Top

Error messages

***ESCAPE Messages**

CPF89A9
Unable to end job that controls directory shadowing.

Top

End Do Group (ENDDO)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The End Do (ENDDO) command is used with the DO command to identify a group of commands that are processed together as a group. The ENDDO command specifies the end of the Do group that is started with an associated DO command. The ENDDO command must be specified after the last command in the Do group.

When Do groups are nested, each group must have its own ENDDO command at its end. Every ENDDO command must be associated with a DO command; if too many ENDDO commands occur in the CL program or ILE CL procedure source, a message is issued and the program is not created.

Restrictions: This command is valid only within a CL program or ILE CL procedure.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

Example 1: Processing a Group of Commands Unconditionally

```
DO
 :   (group of CL commands)
ENDDO
```

The commands between the DO and ENDDO commands are processed once, as a group of commands.

Example 2: Processing a Group of Commands Conditionally

```
IF &SWITCH DO
 :   (group of CL commands)
ENDDO
```

The commands between the DO and ENDDO commands are processed if the value in the logical variable &SWITCH is ’1’. If &SWITCH is not ’1’, then control passes immediately to the next command following the ENDDO command.

Top

Error messages

None

Top

End Disk Reorganization (ENDDSKRGZ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Disk Reorganization (ENDDSKRGZ) command allows the user to end the disk reorganization function started using the Start Disk Reorganization (STRDSKRGZ) CL command. The user can select to end disk reorganization for all auxiliary storage pools (ASPs) or for one or more specific ASPs. A message is sent to the system history (QHST) log when the reorganization function is ended for each ASP.

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

Top

Parameters

Keyword	Description	Choices	Notes
ASP	ASP number	Single values: *ALL Other values (up to 32 repetitions): 1-32	Optional, Positional 1
ASPDEV	ASP device	Values (up to 32 repetitions): *Name*, *ALLAVL	Optional

Top

Auxiliary storage pool ID (ASP)

Specifies for which auxiliary storage pools the disk reorganization function is to be ended. A value must be specified for the ASP parameter or the ASPDEV parameter.

***ALL** Disk reorganization will be ended 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 disk reorganization is to be ended. Valid ASP numbers are 1 to 32. Up to 32 ASP numbers may be specified.

Top

ASP device (ASPDEV)

Specifies the name of the auxiliary storage pool (ASP) device for which the disk reorganization is to be ended. A value must be specified for the ASP parameter or the ASPDEV parameter.

***ALLAVL**
Disk reorganization will be ended 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 disk reorganization is to be ended. Up to 32 ASP device names may be specified.

Top

Examples

Example 1: Ending Disk Reorganization for ASP 1

```
ENDDSKRGZ   ASP(1)
```

This command allows the user to end the disk reorganization function for ASP 1.

Example 2: Ending Disk Reorganization for All ASPs

```
ENDDSKRGZ   ASP(*ALL)
```

This command allows the user to end the reorganization function for each ASP that is currently being reorganized.

Example 3: Ending Disk Reorganization for All ASP Devices

```
ENDDSKRGZ   ASPDEV(*ALLAVL)
```

This command allows the user to end the reorganization function for each ASP device that is currently being reorganized.

Top

Error messages

***ESCAPE Messages**

CPF1889
Disk reorganization not active for ASP &1

CPF1890
*ALLOBJ authority required for requested operation.

Top

End Disk Watcher (ENDDW)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Disk Watcher (ENDDW) command ends a Disk Watcher collection.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Disk Watcher function of the Operating System through System i5 Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_DISK_WATCHER, can also be used to change the list of users that are allowed to use this command.

Top

Parameters

Keyword	**Description**	**Choices**	**Notes**
COL	Collection	*Name*, ***SELECT**	Optional
LIB	Library	*Name*	Optional

Top

Collection (COL)

Specifies the Disk Watcher collection to end.

***SELECT**
A list of active Disk Watcher collections will be displayed for user selection. This list will also identify the job in which each collection is running. This value is valid only if the command is run in an interactive job.

name Specify the name of the collection which will be ended. The collection name is the same as the name of the member where data is being written.

Top

Library (LIB)

Specifies the library of the Disk Watcher collection to end.

Note: A value must be provided for this parameter if a collection name is specified on the **Collection (COL)** parameter.

name Specify the name of the library where data is being written for this collection.

Top

Examples

Example 1: End a Disk Watcher Collection

```
ENDDW   COL(MYMBR) LIB(MYLIB)
```

This command will end the Disk Watcher collection which is writing data to member MYMBR in the Disk Watcher database files in library MYLIB.

Example 2: Select a Disk Watcher Collection to End from a List

```
ENDDW   COL(*SELECT)
```

This command will display a list of active Disk Watcher collections from which the user can select the collection to end.

Top

Error messages

***ESCAPE Messages**

CPF2401
Not authorized to library &1.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

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

CPFB513
The user does not have the required authority.

CPFB51B
Option *SELECT is only valid if the command is being run interactively.

Top

End EPM Environments (ENDEPMENV)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End EPM Environment (ENDEPMENV) command is used with extended program model (EPM) languages to delete a user-controlled environment. The C/400*, FORTRAN/400*, and Pascal languages are part of the extended program model.

You can use this command to delete a run-time environment for an EPM language application that you created with the STREPMENV command. See the Extended Program Model User's Guide and Reference for more detailed information on the EPM and this command.

Error messages for ENDEPMENV

None

Top

Parameters

Keyword	Description	Choices	Notes
EPMENV	Environment Name	*Character value*, *INACT, *MAIN, *REENT	Required, Positional 1
ENVNBR	Environment Number	1-65535	Optional, Positional 2

Top

Environment Name (EPMENV)

Specifies which user-controlled environment is to be deleted. You must specify one of the special values (*INACT, *REENT, or *MAIN) or an environment name. There is no default for this parameter.

***INACT**
All EPM user-controlled environments that are not invoked are deleted.

***REENT**
The reentrant environment with the corresponding ENVNBR is deleted. You must specify a value for ENVNBR if you specify this parameter.

***MAIN**
The *MAIN user-controlled environment is deleted.

environment-name
Enter the name of the environment that you want to delete. The environment name must not begin with an asterisk (*).

Top

Environment Number (ENVNBR)

Specifies the environment number for the *REENT environment that is to be deleted. This parameter is only valid if *REENT is specified on EPMENV.

environment-number
Enter the number of the reentrant environment.

Top

Examples

None

Top

Error messages

None

Top

End Group Job (ENDGRPJOB)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The End Group Job (ENDGRPJOB) command ends a single job within a group and resumes another job within the group. You can specify the following:

- Which job in the group is ended
- Which job in the group gains control (this is valid only when a job is ending itself)
- Whether a job log is created for the job being ended

Top

Parameters

Keyword	Description	Choices	Notes
GRPJOB	Group job	*Name*, *****	Optional, Positional 1
RSMGRPJOB	Group job to be resumed	*Name*, ***PRV**	Optional
LOG	Job log	***NOLIST**, *LIST	Optional

Top

Group job (GRPJOB)

Specifies the group job name of the job being ended.

***** The group job that issued this command is ended.

name Specify the group job name of the job being ended.

Top

Group job to be resumed (RSMGRPJOB)

Specifies the group job name of the job that is resumed after the active job in the group has ended. This parameter is valid only when the job that issues this command is ending itself.

***PRV** The most recently active group job is resumed.

name Specify the group job name of the job that is resumed after the active job in the group ends.

Top

Job log (LOG)

Specifies whether to produce the job log for the ending group job.

***NOLIST**
The information in the job log is not spooled to an output queue.

***LIST** The information in the job log is spooled to an output queue.

Top

Examples

Example 1: Ending Group Job that Issued Command

```
ENDGRPJOB   GRPJOB(*)  LOG(*LIST)  RSMGRPJOB(GROUPJOB1)
```

This command ends the job that is currently running. Its job log is spooled to an output file for printing. When the job completes running, group job GROUPJOB1 becomes the active job in the group.

Example 2: Printing Output of Ended Job

```
ENDGRPJOB   GRPJOB(GROUPJOB2)  LOG(*LIST)
```

Assume that the job issuing the ENDGRPJOB command is group job GROUPJOB1, which wants to end GROUPJOB2. Group job GROUPJOB2 ends. Its job log is spooled to an output file for printing.

Example 3: Ending a Job That's Part of a Secondary Job Pair

```
ENDGRPJOB   GRPJOB(*)  LOG(*NOLIST)
```

Assume that the job issuing the ENDGRPJOB command is the only job in the group and is part of a secondary job pair. The job issuing the command ends. The job's job log is not spooled to an output file. When the job ends, the other job in the secondary job pair is resumed.

Top

Error messages

***ESCAPE Messages**

CPF1309
Subsystem cannot complete the &1 command.

CPF1314
Value &1 for parameter &2 not allowed.

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1322
The End Group Job command not allowed at this time.

CPF1323
Group job &1 not ended; parameters do not agree.

CPF1324
Group job &1 not ended; parameters do not agree.

CPF1325
Group job &1 not ended; group job &2 does not exist.

CPF1326
Group job &1 does not exist.

CPF1327
Cannot end group job &1 with ENDGRPJOB.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

Top

End Host Server (ENDHOSTSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Host Server (ENDHOSTSVR) command is used to end the optimized host server daemons. One or more server daemons can be ended and the server mapper daemon can be ended. Optionally, active connections to the *DATABASE and *FILE servers can be ended with this command.

If a server daemon is ended, and there are servers of that type that have active connections to client applications, the server jobs will remain active until communication with the client application is ended, unless the optional ENDACTCNN parameter is specified. Subsequent connection requests from the client application to that server daemon will fail however until the server daemon is started again.

If the server mapper daemon is ended, any existing client connections to the server jobs are unaffected. Subsequent requests from a client application to connect to the server mapper daemon (to obtain a server's port number) will fail however until the server mapper is started again.

A request to end *ALL host server daemons will end any active daemons.

The ENDACTCNN parameter may be specified in order to end active connections to the *DATABASE and *FILE servers. This will cause the server jobs which are servicing these connections to be ended. The active connections can only be ended if the corresponding daemon job is also being ended. If the *DATABASE keyword is specified, the QZDASOINIT and QZDASSINIT jobs which have active connections will be ended. If the *FILE keyword is specified, the QPWFSERVSO and QPWFSERVSS jobs which have active connections will be ended.

Error messages for ENDHOSTSVR

None

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server type	Single values: *ALL Other values (up to 8 repetitions): *CENTRAL, *DATABASE, *DTAQ, *FILE, *NETPRT, *RMTCMD, *SIGNON, *SVRMAP	Required, Positional 1
ENDACTCNN	End active connections	Single values: *NONE Other values (up to 2 repetitions): *DATABASE, *FILE	Optional, Positional 2

Top

Server type (SERVER)

Specifies the server daemons to be ended.

The possible values are:

***ALL** All of the server daemons and the server mapper daemon are ended.

***CENTRAL**
The central server daemon in the QSYSWRK subsystem, if active, is ended.

***DATABASE**
The database server daemon in the QSERVER subsystem, if active, is ended.

***DTAQ**
The data queue server daemon in the QSYSWRK subsystem, if active, is ended.

***FILE** The file server daemon in the QSERVER subsystem, if active, is ended.

***NETPRT**
The network print server daemon in the QSYSWRK subsystem, if active, is ended.

***RMTCMD**
The remote command and distributed program call server daemon in the QSYSWRK subsystem, if active, is ended.

***SIGNON**
The signon server daemon in the QSYSWRK subsystem, if active, is ended.

***SVRMAP**
The server mapper daemon in the QSYSWRK subsystem, if active, is ended.

Top

End active connections (ENDACTCNN)

Specifies whether or not the active connections for the specified servers will be ended.

Single Value

***NONE:**
No active connections will be ended.

Specific Server Values

***DATABASE:**
The active connections being serviced by the QZDASOINIT and QZDASSINIT server jobs will be ended. The server jobs servicing these connections will be ended.

***FILE:** The active connections being serviced by the QPWFSERVSO and QPWFSERVSS server jobs will be ended. The server jobs servicing these connections will be ended.

Top

Examples

None

Top

Error messages

None

Top

End Input (ENDINP)

Where allowed to run:

- Batch job (*BATCH)

Threadsafe: No

Parameters
Examples
Error messages

The End Input (ENDINP) command is a delimiter in a batch input stream that indicates the end of the input data. The End Input (ENDINP) command also can indicate the end of an inline data file provided the command is detected while the inline file is being processed. If the inline file is using ending characters which are not defaults (//) the End Input (ENDINP) command is embedded without being recognized.

Restrictions: This command cannot be entered at a work station. Two slashes (//) in positions 1 and 2 must go before the End Input (ENDINP) command in the data record, for example //ENDINP. Blanks can separate the slashes from the command line (// ENDINP).

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
//BCHJOB
  :
//DATA
  :
//ENDINP
```

The ENDINP command indicates the end of a input stream that began with the Batch Job (BCHJOB) command.

Top

Error messages

***ESCAPE Messages**

CPF1753
Command cannot be run.

Top

End IP over SNA Interface (ENDIPSIFC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End IP over SNA Interface (ENDIPSIFC) command is used to end an AF_INET sockets over SNA interface (an IP address by which this local host is known on the SNA transport).

Note: Ending an interface causes all routes associated with this interface to be deactivated immediately unless there are other active interfaces that the routes can switch to.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*	Required, Positional 1

Top

Internet address (INTNETADR)

Specifies the internet address of an active (started) interface that had previously been added to the IP SNA configuration with the Add IP over SNA Interface (ADDIPSIFC) CL command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

Examples

```
ENDIPSIFC   INTNETADR('9.5.1.248')
```

This command deactivates (ends) the interface with IP address 9.5.1.248.

Top

Error messages

*ESCAPE Messages

CPFA114
IP over SNA interface &1 not ended.

Top

End Job (ENDJOB)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Job (ENDJOB) command ends the specified job and any associated inline data files. The job can be on a job queue, it can be active, or it can have already completed running.

You can specify that the application program is given time to control end-of-job processing. If no time is given or if cleanup cannot be performed within the given time, the system performs minimal end-of-job processing, which can include:

- Closing the database files.
- Spooling the job log to an output queue.
- Cleaning up internal objects in the operating system.
- Showing the end-of-job display (for interactive jobs).
- Completing commitment control processing

  Before ending the job, you should verify that no logical unit of work is in an in doubt state due to a two-phase commit operation that is in progress. If it is, then the value of the Action if ENDJOB commitment option can greatly impact the ENDJOB processing. For example, if the Action if ENDJOB commitment option is the default value of WAIT, this job will be held up and will not complete its end of job processing until the commitment control operation is completed. This ensures database integrity on all related systems. For specific instructions on how to determine these conditions, and for a description of all the impacts of ending this job under these conditions, see the Backup and Recovery book.

Restrictions: The issuer of the command must be running under a user profile which is the same as the job user identity of the job being ended, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. This restriction is enforced even when ending the current job.

The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management book.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	Single values: * Other values: *Qualified job name*	Required, Positional 1
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 2
DELAY	Controlled end delay time	1-999999, **30**	Optional, Positional 3
SPLFILE	Delete spooled files	***NO**, *YES	Optional, Positional 4

Keyword	Description	Choices	Notes
LOGLMT	Maximum log entries	*Integer*, ***SAME**, *NOMAX	Optional
ADLINTJOBS	Additional interactive jobs	***NONE**, *GRPJOB, *ALL	Optional
DUPJOBOPT	Duplicate job option	***SELECT**, *MSG	Optional

Top

Job name (JOB)

Specifies the qualified job name of the job to be ended.

This is a required parameter.

Single values

* The job from which this command is run is ended. Specifying this value is the only way to end the current job.

Qualifier 1: Job name

name Specify the name of the job.

Qualifier 2: User

name Specify the user name that identifies the user profile under which the job is run.

Qualifier 3: Number

000000-999999
Specify the system-assigned job number.

Note: If no user name or job number is specified, all jobs currently in the system are searched for the job name. If more than one occurrence of the specified name is found, a qualified job name must be provided either explicitly or through the selection display. Refer to the **Duplicate job option (DUPJOBOPT)** parameter for more information.

Top

How to end (OPTION)

Specifies whether the job ends immediately or in a controlled manner that lets the application program perform end-of-job processing. In either case, the system performs certain job cleanup processing.

***CNTRLD**
The job ends in a controlled manner. This allows the program running to perform cleanup (end-of-job processing). When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. The application has the amount of time specified on the DELAY parameter to complete cleanup before the job is ended.

***IMMED**
The job ends immediately and the system performs end-of-job cleanup. System cleanup can take from a brief amount of time to several minutes. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job and the QENDJOBLMT system value specifies the time limit. Other than by handling the SIGTERM signal, the program that is running is not allowed to perform any cleanup.

Note: The *IMMED value might cause undesirable results if data has been partially updated. This value should be used only after a controlled end has been attempted unsuccessfully.

Note: When a SIGTERM signal handler is running during the immediate ending of a job, an ENDJOB command with OPTION(*IMMED) can be used to end the SIGTERM signal handler. This is only allowed if the SIGTERM signal handler has already had at least two minutes to run.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed for the job to complete its cleanup processing during a controlled end. If the cleanup is not completed before the end of the delay time, the job is ended immediately. (Only system cleanup is performed.)

The delay time does not start until the job becomes active if the job is suspended because of one of the following conditions:

- The system request option 1 is selected.
- The job is held by the Hold Job (HLDJOB) command.
- The job is transferred by the Transfer Secondary Job (TFRSECJOB) command.
- The job is transferred by the Transfer to Group Job (TFRGRPJOB) command.

Note: This parameter is valid only when OPTION(*CNTRLD) is specified.

30 A maximum delay time of 30 seconds is allowed for cleanup before the job ends.

1-999999
Specify the maximum amount of delay time (in seconds) before the job ends.

Top

Delete spooled files (SPLFILE)

Specifies whether spooled output files created by this job are kept for normal processing or deleted. Regardless of whether the spooled files are deleted, the job logs are kept.

***NO** The spooled output files created by the job being ended are kept for normal processing by a writer. When the job ends, the spooled file action (SPLFACN) job attribute determines whether spooled files are detached from the job or kept with the job.

***YES** The spooled output files created by the job being ended and which are on output queues in the library name space of the thread issuing this command are deleted. The job log is not deleted. If the job has already ended and the spooled file action for the job is to detach the spooled files, the End Job (ENDJOB) command will not find the job and the spooled files will not be deleted.

Top

Maximum log entries (LOGLMT)

Specifies the maximum number of entries in the message queue of the job being ended that are written to the job log. This parameter can be used to limit the number of messages written to the job log printer file, QPJOBLOG, for a job that ends.

The value specified on this parameter can change the logging limit of the job even if the job is already ending or the job has already ended. The following are examples of how the logging limit can be changed:

1. If the value specified is greater than the number of messages written at the time the command is issued, messages continue to be written until the new limit is reached.
2. If the value specified is less than the number of messages already written to the spooled file, a message indicating that the limit has been reached is immediately put in the spooled file as the last entry. The remaining messages on the queue are ignored.
3. If 0 (zero) is specified before any messages are written to the spooled file, no job log is produced for the job that is ending. If the job has already ended and the job log has not yet been produced, the job log is removed regardless of the value of the job log output (LOGOUTPUT) job attribute. For more information on removing pending job logs, refer to the Remove Pending Job Log (QWTRMVJL) API.

***SAME**
The message logging limit does not change. If the logging limit does not change for this job on a previous command, *NOMAX is the value used by the system.

***NOMAX**
There is no limit to the number of messages logged; **all** messages on the job message queue are written to the job log.

integer-number
Specify the maximum number of messages that can be written to the job log.

Top

Additional interactive jobs (ADLINTJOBS)

Specifies whether the additional interactive jobs associated with the job specified in the **Job name (JOB)** parameter are ended.

***NONE**
Only the job specified in the JOB parameter is ended.

***GRPJOB**
If the job specified in the JOB parameter is a group job, all group jobs associated with the group are ended. If the job is not a group job, the job specified in the JOB parameter is ended.

***ALL** All interactive jobs running on the workstation associated with the job specified in the JOB parameter are ended. This includes group jobs and secondary jobs.

Top

Duplicate job option (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. Otherwise, a message is issued.

***MSG** A message is issued when duplicate jobs are found.

Top

Examples

Example 1: Ending a Job Immediately

```
ENDJOB   JOB(JOB1)  OPTION(*IMMED)  SPLFILE(*YES)
```

This command ends a job named JOB1 immediately. Spooled output produced by the job is deleted; the job log is saved.

Example 2: Saving Spooled Output

```
ENDJOB   JOB(001234/XYZ/JOB2)  OPTION(*CNTRLD)
         DELAY(50)  SPLFILE(*NO)
```

This command ends a job named 001234/XYZ/JOB2. Spooled output is saved for normal processing by the spooling writer. The job has 50 seconds to perform any cleanup routines, after which it is ended immediately.

Example 3: Removing a Pending Job Log

```
ENDJOB   JOB(543210/ABCDE/JOB3)  LOGLMT(0)
```

This command removes the pending job log for a completed job named 543210/ABCDE/JOB3. This has the same effect as the Remove Pending Job Log (QWTRMVJL) API.

Example 4: Ending the Current Job

```
ENDJOB   JOB(*)  OPTION(*IMMED)
```

This command ends the job in which the command is issued. To exit from a SIGTERM signal handling procedure after the required cleanup has been performed, immediately end the current job. Control may be returned to the command issuer, even though the immediate option has been used. To handle this situation, the program which issued this command should return.

Top

Error messages

***ESCAPE Messages**

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1321
Job &1 user &2 job number &3 not found.

CPF1332
End of duplicate job names.

CPF1340
Job control function not performed.

CPF1341
Reader or writer &3/&2/&1 not allowed as job name.

CPF1342
Current job not allowed as job name on this command.

CPF1343
Job &3/&2/&1 not valid job type for function.

CPF1344
Not authorized to control job &3/&2/&1.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1352
Function not done. &3/&2/&1 in transition condition.

CPF135D
ENDJOB OPTION(*IMMED) not allowed at this time.

CPF1360
&3/&2/&1 already ending because of ENDJOBABN.

CPF1361
Job &3/&2/&1 already ending with *IMMED option.

CPF1362
Job &3/&2/&1 has completed.

CPF1363
Job &3/&2/&1 is already ending *CNTRLD.

CPF8172
Spool control block for job &10/&9/&8 damaged.

Top

End Job Abnormal (ENDJOBABN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Job Abnormal (ENDJOBABN) command ends a job that cannot be ended successfully by running the End Job (ENDJOB) command with *IMMED specified for the **How to end (OPTION)** parameter. The ENDJOBABN command cannot be issued against a job until 10 minutes have passed following the request for immediate ending. This allows sufficient time for normal job ending functions to be attempted.

When the ENDJOBABN command is issued, most of the end-of-job processing is bypassed (including spooling of the job log, the end of job display for interactive jobs, and the end-of-job processing for the specific functions that are being performed). The part of the end-of-job processing that is attempted is allowed only five minutes to complete. If it does not do so in five minutes, the job is forced to end at that point. Because some of the job cleanup is not performed, the ENDJOBABN command should only be used when a job that is in the process of immediate ending does not finish ending and resources in use by the job are needed by another job or by the system. When the ENDJOBABN command is used, some resources in use by the ended job may be left unavailable until the next IPL.

Use of the ENDJOBABN command causes the next system end to be marked as ABNORMAL. Certain system functions are then called during the subsequent IPL to clear up conditions that may have resulted from running the ENDJOBABN command. This does not, however, cause any machine recovery functions to be called, nor do any access paths need to be rebuilt. Some storage in use by the job may become unavailable after the ENDJOBABN command is run and that available storage can be reclaimed by using the Reclaim Storage (RCLSTG) command.

Bypassing the job log writing process causes the job to have the status of JOBLOG PENDING (as shown on the DSPJOB status attributes display) after it has been ended with the ENDJOBABN command. The job log writing is not performed until the next IPL. However, the contents of the job log can be printed or shown by using the Display Job Log (DSPJOBLOG) command.

When the ENDJOBABN command is run, the following functions are performed successfully:

- Journaling entries
- Commitment control

  Before ending the job abnormally, you should verify that no logical unit of work is in an in doubt state due to a two-phase commit operation that is in progress. If it is, then pending committable changes at this system will not be committed or rolled back. Therefore, database integrity may not be maintained on all related systems. For specific instructions on how to determine these conditions, and for a description of all the impacts of ending this job abnormally under these conditions, see the Commitment control article in the Information Center.
- Making database files available for use by others
- Releasing file locks

This command fails to end a job or takes more than five minutes to do so in the following situations:

- When the job runs under a subsystem monitor that is hung, is abnormally slow, or has ended abnormally (the subsystem monitor performs part of the ending function).
- When the machine interface (MI) instruction running in the job is hung or is abnormally slow. The job cannot end until the MI instruction that is currently running completes or reaches a point of interruption.

Restrictions:

1. The issuer of the command must be running under a user profile which is the same as the job user identity of the job being ended, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management book.
2. After the ENDJOBABN command is run, subsequent ENDJOBABN commands cannot be issued against the job.
3. Users cannot end a reader, writer, subsystem monitor, or system job.
4. Users cannot run the ENDJOBABN command until ten minutes after immediate ending of the job is started. Immediate ending of the job is started in the following ways:
   - When the End Job (ENDJOB) command with OPTION(*CNTRLD) is specified and the delay time ends.
   - When the ENDJOB command with OPTION(*IMMED) is issued.
   - When the End Subsystem (ENDSBS) command with OPTION(*CNTRLD) is issued against the subsystem in which the job is running and the delay time ends.
   - When the ENDSBS command with OPTION(*IMMED) is issued against the subsystem in which the job is running.
   - When the End System (ENDSYS) command with OPTION(*IMMED) is issued, or OPTION(*CNTRLD) is issued and the delay time ends.
   - When the Power Down System (PWRDWNSYS) command with OPTION(*IMMED) is issued.
5. If the job defines a handler for the asynchronous signal SIGTERM, the immediate ending of the job was delayed to allow the SIGTERM signal handler to run. For more information, refer to system value QENDJOBLMT. An ENDJOBABN command is not allowed while the SIGTERM signal handler is running. If the SIGTERM signal handler has run for at least 2 minutes, use ENDJOB command with OPTION(*IMMED) to stop the SIGTERM signal handler.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	*Qualified job name*	Required, Positional 1
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
DUPJOBOPT	Duplicate job option	***SELECT**, *MSG	Optional

Top

Job name (JOB)

Specifies the qualified job name of the job to be ended.

This is a required parameter.

Qualifier 1: Job name

name Specify the name of the job.

Qualifier 2: User

name Specify the user name that identifies the user profile under which the job is run.

Qualifier 3: Number

000000-999999
Specify the system-assigned job number.

Note: If no user name or job number is specified, all jobs currently in the system are searched for the job name. If more than one occurrence of the specified name is found, a qualified job name must be provided either explicitly or through the selection display. Refer to the **Duplicate job option (DUPJOBOPT)** parameter for more information.

Top

Duplicate job option (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. Otherwise, a message is issued.

***MSG** A message is issued when duplicate jobs are found.

Top

Examples

```
ENDJOBABN   JOB(000310/SMITH/PAYROLL)
```

This command ends the batch job 000310/SMITH/PAYROLL after the failure of an earlier attempt to end it with the ENDJOB command. The ENDJOBABN command can be issued only after waiting at least ten minutes for the job to end after issuing the ENDJOB command.

Top

Error messages

***ESCAPE Messages**

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1321
Job &1 user &2 job number &3 not found.

CPF1332
End of duplicate job names.

CPF1340
Job control function not performed.

CPF1341
Reader or writer &3/&2/&1 not allowed as job name.

CPF1342
Current job not allowed as job name on this command.

CPF1343
Job &3/&2/&1 not valid job type for function.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1359
ENDJOBABN not allowed at this time for job &3/&2/&1.

CPF1360
&3/&2/&1 already ending because of ENDJOBABN.

CPF1362
Job &3/&2/&1 has completed.

Top

End Journal (ENDJRN)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The End Journal (ENDJRN) command is used to end the journaling of changes for an object or list of objects. The object types which are supported through this interface are Data Areas (*DTAARA), Data Queues (*DTAQ), Stream Files (*STMF), Directories (*DIR), and Symbolic Links (*SYMLNK). Only objects of type *STMF, *DIR, or *SYMLNK that are in the ″root″ (/), QOpenSys, and user-defined file systems are supported.

All objects of the types supported by this command that are currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:

1. Access Paths - End Journal Access Path (ENDJRNAP)
2. Physical Files - End Journal Physical File (ENDJRNPF)
3. Libraries - End Journal Library (ENDJRNLIB)
4. Other Objects - End Journal Object (ENDJRNOBJ)

Restrictions:

- If OBJ(*ALL) is specified, a journal name must be specified (JRN parameter).
- If a journal name and a list of object names are specified, all objects must be currently journaled to the indicated journal.
- The specified journal must be a local journal.
- At least one of parameter OBJ or OBJFID must be specified.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): *Element list*	Optional
	Element 1: Name	*Path name*, *	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
OBJFID	File identifier	Values (up to 300 repetitions): *Hexadecimal value*	Optional
SUBTREE	Directory subtree	***NONE**, *ALL	Optional
PATTERN	Name pattern	Values (up to 20 repetitions): *Element list*	Optional
	Element 1: Pattern	*Character value*, *	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
JRN	Journal	*Path name*, ***OBJ**	Optional
LOGLVL	Logging level	***ERRORS**, *ALL	Optional

Top

Objects (OBJ)

Specifies a maximum of 300 object path names for which changes will no longer be journaled. Only objects whose path name identifies an object of type *STMF, *DIR, *SYMLNK, *DTAARA or *DTAQ are supported.

Single values

***ALL** All objects of the supported type that are currently being journaled to the indicated journal are to stop having their changes journaled. If *ALL is specified parameter OBJFID must not be specified.

Element 1: Name

'object-path-name'
Specify the path name of the object for which changes are no longer journaled.

A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. Symbolic links within the path name will not be followed. If the path name begins with the tilde character, then the path is assumed to be relative to the appropriate home directory.

Additional information about path name patterns is in the Integrated file system topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Element 2: Include or omit

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

***INCLUDE**
The objects that match the object name pattern are to stop having their changes journaled unless overridden by an *OMIT specification.

***OMIT**
The objects that match the object name pattern are not to be included with the objects that are to stop having their changes journaled. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected path.

Top

File identifier (OBJFID)

Specifies a maximum of 300 file identifiers (FID) for which changes are no longer journaled. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies an object of type *STMF, *DIR, or *SYMLNK that is in the ″root″ (/), QOpenSys, or user-defined file systems, or objects of type *DTAARA or *DTAQ are supported.

file-identifier
Objects identified with the FID are no longer journaled.

Top

Directory subtree (SUBTREE)

Specifies whether the objects in directory subtrees are to stop having their changes journaled.

Note: This parameter is ignored unless object-path-name is a directory (*DIR) object.

Note: This parameter is ignored if the OBJFID parameter is specified.

***NONE**
Only the objects that match the selection criteria are processed. The objects within selected directories are not implicitly processed.

***ALL** All objects that meet the selection criteria are processed in addition to the entire subtree of each directory that matches the selection criteria. The subtree includes all subdirectories and the objects within those subdirectories.

Once the command has begun processing a specific directory subtree, the objects which will be found and processed may be affected by operations that update the organization of objects within the specified directory tree. This includes, but is not limited to, the following:

- Adding, removing, or renaming object links
- Mounting or unmounting file systems
- Updating the effective root directory for the process calling the command
- Updating the contents of a symbolic link

In order to process the directory subtree, the system code may increase the process-scoped maximum number of file descriptors that can be opened during processing. This is done so that the command is not likely to fail due to a lack of descriptors. This process-scoped maximum value is not reset when the command completes.

Top

Name pattern (PATTERN)

Specifies a maximum of 20 patterns to be used to include or omit objects for the end journal operation.

Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. Symbolic links within the path name will not be followed.

If this parameter is not specified, the default will be to match all patterns.

Additional information about path name patterns is in the Integrated file system topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Note: This parameter is ignored if the OBJFID parameter is specified.

Element 1: Name pattern

′*′ All objects that match the input OBJ parameter are to be included into the end journal operation or omitted from the end journal operation.

name-pattern
Specify the pattern to either include or omit objects for the end journal operation. Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern.

Element 2: Include or omit

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

***INCLUDE**
The objects that match the object name pattern are to stop having their changes journaled unless overridden by an *OMIT specification.

***OMIT**
The objects that match the object name pattern are not to be included with the objects that are to stop having their changes journaled. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Journal (JRN)

Specifies the journal to which changes are currently being journaled.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

***OBJ** The journal is determined by the system from the specified object path name or object file identifier.

journal-path-name
Specify the path name of the journal to which changes are currently being journaled.

Top

Logging level (LOGLVL)

Specifies the error logging level used. This parameter is used to determine which messages will be sent.

***ERRORS**
All diagnostic and escape messages are sent but the command will not send successful completion messages for each object. At the completion of this command, one completion message will be sent.

***ALL** The command sends all the messages that would be sent with *ERRORS and it will also send the successful completion message for each object.

Top

Examples

Example 1: End All Non-Database Journaling

```
ENDJRN   OBJ(*ALL)  JRN('/qsys.lib/mylib.lib/myjrn.jrn')
```

This command stops the journaling of all changes to all objects of type *DIR, *STMF, *SYMLNK, *DTAARA and *DTAQ to journal /qsys.lib/mylib.lib/myjrn.jrn.

Example 2: End Journaling with Omit of Directory

```
ENDJRN   OBJ(('/mypath' *INCLUDE) ('/mypath/myobject' *OMIT))
```

This command stops the journaling of all changes to all first-level objects in directory /mypath except object /mypath/myobject. Object /mypath/myobject will continue to be journaled.

Example 3: End Journaling with Pattern Selection

```
ENDJRN   OBJ(('/mypath' *INCLUDE) ('/mypath/mysubdir' *OMIT))
         SUBTREE(*ALL)  PATTERN(('*.txt' *INCLUDE))
```

This command stops the journaling of all changes to all objects in directory /mypath of type *DIR, *STMF, and *SYMLNK that match pattern '*.txt'. Any objects within directory /mypath/mysubdir will continue to be journaled.

Example 4: End Journaling using File Identifiers

```
ENDJRN   OBJFID(00000000000000007E09BDB000000009
                00000000000000009E09BDB00000000A)
```

This command stops the journaling of all changes to the objects of type *DIR, *STMF, *SYMLNK, *DTAARA or *DTAQ represented by the specified file identifiers.

Example 5: End Journaling on a set of Data Areas

```
ENDJRN   OBJ(('/qsys.lib/mylib.lib/mydata*.dtaara'))
         JRN('/qsys.lib/mylib.lib/myjrn.jrn')
```

This command stops the journaling of all changes to the objects of type *DTAARA in library MYLIB that begin with the characters 'MYDATA'.

Top

Error messages

***ESCAPE Messages**

CPFA0D4
File system error occurred. Error number &1.

CPF700B
&1 of &2 objects have ended journaling.

CPF705A
Operation failed due to remote journal.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

End Journal Access Path (ENDJRNAP)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The End Journal Access Path (ENDJRNAP) command is used to end the journaling of the access paths of a journaled file.

All access paths currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:
- Integrated file system objects - End Journal (ENDJRN)
- Physical files - End Journal Physical File (ENDJRNPF)
- Libraries - End Journal Library (ENDJRNLIB)
- Other objects - End Journal Object (ENDJRNOBJ)

Restrictions:
- Overrides are not applied to the files listed in the FILE parameter.
- If FILE(*ALL) is specified, a journal name must be specified.
- If a journal name and a list of file names are specified, then all the access paths for the listed files must be currently journaled to the indicated journal.
- Journaling entries for any physical file does not end by the running of this command.
- The specified journal must be a local journal.
- Lock Processing

  The file to end journaling is locked with a read exclusive lock (*EXCLRD).

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Journaled file	Single values: *ALL Other values (up to 300 repetitions): *Qualified object name*	Required, Positional 1
	Qualifier 1: Journaled file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JRN	Journal	Single values: ***FILE** Other values: *Qualified object name*	Optional, Positional 2
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LOGLVL	Logging level	***ERRORS**, *ALL	Optional

Top

Journaled file (FILE)

Specifies a maximum of 300 database files for which the journaling of access paths are to be ended.

This is a required parameter.

Single values

***ALL** All current journaling of access paths to the indicated journal are ended.

Qualifier 1: Journaled file

file-name
Specify the name and library of the database file for which access paths for the journal entry are ended.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Journal (JRN)

Specifies the journal to which journaling of the access paths for the indicated files are being ended.

Single values

***FILE** The journal name is determined by the system from the specified file names.

Qualifier 1: Journal

journal-name
Specify the name of the journal.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Logging level (LOGLVL)

Specifies the error logging level used. This parameter is used to determine which messages will be sent.

***ERRORS**
All diagnostic and escape messages are sent but the command will not send successful completion messages for each object. At the completion of this command, one completion message will be sent.

***ALL** The command sends all the messages that would be sent with *ERRORS and it will also send the successful completion message for each object.

Top

Examples

```
ENDJRNAP   FILE(MYLIB/MYFILE)
```

This command ends the journaling for all access paths of the file MYFILE in the library MYLIB.

Top

Error messages

***ESCAPE Messages**

CPF6972
Cannot allocate access path for file &1 in &2.

CPF7008
Cannot start or end access path journaling for file &1.

CPF703C
DDL transaction prevents journaling operation.

CPF703D
DDL transaction prevents journaling operation.

CPF703E
DDL transaction prevents journaling operation.

CPF7032
ENDJRNPF or ENDJRNAP command failed.

CPF7033
Start or end journaling failed for member &3.

CPF7034
Logical damage of file &1 in &2.

CPF705A
Operation failed due to remote journal.

CPF708D
Journal receiver found logically damaged.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

End Journal Library (ENDJRNLIB)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End Journal Library (ENDJRNLIB) command is used to end journaling of changes for a library or list of libraries.

All libraries that are currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:

- Access paths - End Journal Access Path (ENDJRNAP)
- Integrated file system objects - End Journal (ENDJRN)
- Physical files - End Journal Physical File (ENDJRNPF)
- Other objects - End Journal Object (ENDJRNOBJ)

Restrictions:

- If LIB(*ALL) is specified, a journal name must be specified (JRN parameter).
- If a journal name and a list of library names are specified, all objects must be currently journaled to the indicated journal.
- The specified journal must be a local journal. RCVSIZOPT(*MAXOPT3) specified.
- Lock Processing

  The library to end journaling is locked with a read exclusive lock (*EXCLRD).

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Single values: *ALL Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
JRN	Journal	Single values: ***OBJ** Other values: *Qualified object name*	Optional, Positional 2
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LOGLVL	Logging level	***ERRORS**, *ALL	Optional

Top

Library (LIB)

Specifies a maximum of 300 library names for which changes are no longer to be journaled.

Single values

***ALL** All libraries that are currently being journaled to the indicated journal will no longer have their changes journaled.

Other values

generic-name
Specify the generic name of the library for which changes will no longer be journaled. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all libraries that have names with the same prefix as the generic name will no longer have their changes journaled.

library-name
Specify the name of the library for which changes will no longer be journaled.

Top

Journal (JRN)

Specifies the qualified name of the journal to which changes in the libraries are currently being journaled.

Single values

***OBJ** The journal is determined by the system from the specified library name.

Qualifier 1: Journal

journal-name
Specify the name of the journal to which the indicated libraries are currently being journaled.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Logging level (LOGLVL)

Specifies the error logging level used. This parameter is used to determine which messages will be sent.

***ERRORS**
All diagnostic and escape messages are sent but the command will not send successful completion messages for each object. At the completion of this command, one completion message will be sent.

***ALL** The command sends all the messages that would be sent with *ERRORS and it will also send the successful completion message for each object.

Top

Examples

Example 1: End Journaling All Libraries

```
ENDJRNLIB   LIB(*ALL)  JRN(MYLIB/MYJRN)
```

This command stops journaling all changes to all libraries to journal MYJRN in library MYLIB.

Example 2: End Journaling for Specific Library

```
ENDJRNLIB   LIB(DTALIB)
```

This command stops the journaling of all changes to library DTALIB.

Top

Error messages

***ESCAPE Messages**

CPF700B
&1 of &2 objects have ended journaling.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

Top

End Journal Object (ENDJRNOBJ)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The End Journal Object (ENDJRNOBJ) command is used to end journaling of changes for an object or list of objects.

All objects, of object types *DTAARA and *DTAQ, that are currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:

- Access paths - End Journal Access Path (ENDJRNAP)
- Integrated file system objects - End Journal (ENDJRN)
- Physical files - End Journal Physical File (ENDJRNPF)
- Libraries - End Journal Library (ENDJRNLIB)

Restrictions:

- If OBJ(*ALL) or OBJTYPE(*ALL) is specified, a journal name must be specified (JRN parameter).
- If a journal name and a list of object names are specified, all objects must be currently journaled to the indicated journal.
- The specified journal must be a local journal.
- Lock Processing

  The object to end journaling is locked with a read exclusive lock (*EXCLRD).

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	Single values: *ALL Other values (up to 300 repetitions): *Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Generic name, name*, *ALLLIB	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OBJTYPE	Object type	*DTAARA, *DTAQ, *ALL	Required, Positional 2
JRN	Journal	Single values: ***OBJ** Other values: *Qualified object name*	Optional, Positional 3
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LOGLVL	Logging level	***ERRORS**, *ALL	Optional

Top

Object (OBJ)

Specifies a maximum of 300 objects for which changes are no longer to be journaled.

This is a required parameter.

Single values

***ALL** All objects of the specified object types that are currently being journaled to the indicated journal will no longer have their changes journaled.

Qualifier 1: Object

***ALLLIB**
All objects of the specified object types in the specified library will no longer have their changes journaled.

generic-name
Specify the generic name of the object for which changes will no longer be journaled. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all objects of the specified object type that have names with the same prefix as the generic name will no longer have their changes journaled.

object-name
Specify the name of the object for which changes will no longer be journaled.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Object type (OBJTYPE)

Specifies the object type for which journaling is to be ended.

This is a required parameter.

***DTAARA**
Data area objects are to have their journaling ended.

***DTAQ**
Data queue objects are to have their journaling ended.

***ALL** All objects of the object types that are supported on this command are to have their journaling ended.

Note: If OBJTYPE(*ALL) is specified, then OBJ(*ALL) must also be specified.

Top

Journal (JRN)

Specifies the qualified name of the journal to which changes in the objects are currently being journaled.

Single values

***OBJ** The journal is determined by the system from the specified object name and object type.

Qualifier 1: Journal

journal-name
Specify the name of the journal to which the indicated objects are currently being journaled.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Logging level (LOGLVL)

Specifies the error logging level used. This parameter is used to determine which messages will be sent.

***ERRORS**
All diagnostic and escape messages are sent but the command will not send successful completion messages for each object. At the completion of this command, one completion message will be sent.

***ALL** The command sends all the messages that would be sent with *ERRORS and it will also send the successful completion message for each object.

Top

Examples

Example 1: End Journaling All Data Areas and Data Queues

```
ENDJRNOBJ   OBJ(*ALL)  OBJTYPE(*ALL)  JRN(MYLIB/MYJRN)
```

This command stops journaling all changes to all objects of type *DTAARA and *DTAQ to journal MYJRN in library MYLIB.

Example 2: End Journaling for Specific Data Area

```
ENDJRNOBJ   OBJ(DTALIB/MYDTAARA)  OBJTYPE(*DTAARA)
```

This command stops the journaling of all changes to data area MYDTAARA in library DTALIB.

Top

Error messages

***ESCAPE Messages**

CPF700B
&1 of &2 objects have ended journaling.

CPF705A
Operation failed due to remote journal.

CPF7057
*LIBL not allowed with *ALL or *ALLLIB or a generic name.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

End Journal Physical File (ENDJRNPF)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End Journal Physical File (ENDJRNPF) command is used to end journaling of changes for a specific physical file and all of its members.

All physical files currently being journaled to a specific journal may also have journaling stopped.

When the file for which journaling is ended is a distributed file, an attempt is made to distribute the ENDJRNPF command if journaling was successfully ended locally. Even if the distribution request fails, the local file is not journaled. In addition, if a journal and file name are specified, and the file is distributed, an attempt to distribute the ENDJRNPF request is made even if the file is not journaled locally.

For other ways to end journaling see the following commands:

- Access paths - End Journal Access Path (ENDJRNAP)
- Integrated file system objects - End Journal (ENDJRN)
- Libraries - End Journal Library (ENDJRNLIB)
- Other objects - End Journal Object (ENDJRNOBJ)

Restrictions:

- Overrides are not applied to the files listed in the FILE parameter.
- If FILE(*ALL) is specified, a journal name must be specified.
- If a journal name and a list of file names are specified, all files must be currently journaled to the indicated journal.
- The specified journal must be a local journal.
- In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.
- Lock Processing

  The physical file to end journaling is locked with a read exclusive lock (*EXCLRD). Any logical files built over the physical file are also locked with a read exclusive lock (*EXCLRD).

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Journaled physical file	Single values: *ALL Other values (up to 300 repetitions): *Qualified object name*	Required, Positional 1
	Qualifier 1: Journaled physical file	*Generic name, name*, *ALLLIB	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JRN	Journal	Single values: ***FILE** Other values: *Qualified object name*	Optional, Positional 2
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Keyword	Description	Choices	Notes
LOGLVL	Logging level	***ERRORS**, *ALL	Optional

Top

Journaled physical file (FILE)

Specifies a maximum of 300 physical files for which changes will no longer be journaled.

This is a required parameter.

Single values

***ALL** All physical files currently being journaled to the specified journal no longer have their changes journaled.

Qualifier 1: Journaled physical file

***ALLLIB**
All physical files in the specified library will no longer have their changes journaled.

generic-name
Specify the generic name of the physical file for which changes will no longer be journaled. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all physical files that have names with the same prefix as the generic name will no longer have their changes journaled.

file-name
Specify the name of the physical file for which changes will no longer be journaled.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Journal (JRN)

Specifies the name of the journal to which changes in the indicated files are currently being journaled.

Single values

***FILE** The journal is determined by the system from the specified file names.

Qualifier 1: Journal

journal-name
Specify the name of the journal to which changes in the specified files are currently being journaled.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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, QGPL is used.

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

Top

Logging level (LOGLVL)

Specifies the error logging level used. This parameter is used to determine which messages will be sent.

***ERRORS**
All diagnostic and escape messages are sent but the command will not send successful completion messages for each object. At the completion of this command, one completion message will be sent.

***ALL** The command sends all the messages that would be sent with *ERRORS and it will also send the successful completion message for each object.

Top

Examples

```
ENDJRNPF   FILE(MYLIB/MYFILE)
```

This command stops the journaling of all changes to all members of file MYFILE in library MYLIB. Changes made after this command is run are not journaled.

Top

Error messages

***ESCAPE Messages**

CPF700B
&1 of &2 objects have ended journaling.

CPF703C
DDL transaction prevents journaling operation.

CPF703D
DDL transaction prevents journaling operation.

CPF703E
DDL transaction prevents journaling operation.

CPF705A
Operation failed due to remote journal.

CPF7057
*LIBL not allowed with *ALL or *ALLLIB or a generic name.

CPF708D
Journal receiver found logically damaged.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

End Job Watcher (ENDJW)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Job Watcher (ENDJW) command ends a Job Watcher collection.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Job Watcher function of the Operating System through System i5 Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_JOB_WATCHER, can also be used to change the list of users that are allowed to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
COL	Collection	*Name*, ***SELECT**	Optional
LIB	Library	*Name*	Optional

Top

Collection (COL)

Specifies the Job Watcher collection to end.

***SELECT**
A list of active Job Watcher collections will be displayed for user selection. This list will also identify the job in which each collection is running. This value is valid only if the command is run in an interactive job.

name Specify the name of the collection which will be ended. The collection name is the same as the name of the member where data is being written.

Top

Library (LIB)

Specifies the library of the Job Watcher collection to end.

Note: A value must be provided for this parameter if a collection name is specified on the **Collection (COL)** parameter.

name Specify the name of the library where data is being written for this collection.

Top

Examples

Example 1: End a Job Watcher Collection

```
ENDJW   COL(MYMBR) LIB(MYLIB)
```

This command will end the Job Watcher collection which is writing data to member MYMBR in the Job Watcher database files in library MYLIB.

Example 2: Select a Job Watcher Collection to End From a List

```
ENDJW   COL(*SELECT)
```

This command will display a list of active Job Watcher collections from which the user can select the collection to end.

Top

Error messages

***ESCAPE Messages**

CPF2401
Not authorized to library &1.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

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

CPFB518
The user does not have the required authority.

CPFB51B
Option *SELECT is only valid if the command is being run interactively.

Top

End Line Recovery (ENDLINRCY)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Line Recovery (ENDLINRCY) command ends automatic error recovery procedures for a specific line. If any type of failure occurs after this command is run, an inquiry message is sent to the system operator.

Use the Resume Line Recovery (RSMLINRCY) command to reestablish error recovery procedures for the line.

Top

Parameters

Keyword	Description	Choices	Notes
LINE	Line	*Name*	Required, Positional 1

Top

Line (LINE)

Specifies the name of the communications line whose recovery is to be stopped.

This is a required parameter.

Top

Examples

```
ENDLINRCY   LINE(NYC2)
```

This command ends error recovery procedures for the line named NYC2.

Top

Error messages

***ESCAPE Messages**

CPF2704
Line description &1 not found.

CPF5917
Not authorized to line description &1.

CPF5932
Cannot access line &1.

CPF5933
Line &1 not varied on.

CPF5935
Error occurred during command processing.

Top

End Job Log Server (ENDLOGSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Job Log Server (ENDLOGSVR) command is used to end the job log server. The job log server writes job logs for jobs that are in a job log pending state. See Job log output (LOGOUTPUT) for additional information on which jobs are handled by the server. If more than one job log server job is active at the time this command is issued, all of the job log server jobs will be ended.

Restrictions:

- You must have job control (*JOBCTL) special authority to use this command.

Usage notes:

- If you only want to stop the production of a particular job log because, for example, it is very long or consuming too many resources, use the Change Job (CHGJOB) or End Job (ENDJOB) command instead of this command.

  To determine which job a server is currently working on, display the server job's job log using the Display Job Log (DSPJOBLOG) command. Informational message, CPI1307, ″Writing job log for &3/&2/&1″ will be logged in the server job's job log while it is writing the job log for the job named in the message.

  If you want to be able to write the complete job log for the job named in CPI1307 at a later time, change the job using the CHGJOB command specifying *PND on the LOGOUTPUT parameter.

  If you do not need the job log and do not want to save it so it could be rewritten later, use the ENDJOB command on the job named in the CPI1307 message, specifying LOGLMT(0).

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 1

Top

How to end (OPTION)

Specifies whether the job log server is ended in a controlled manner or immediately.

***CNTRLD**
The server is ended in a controlled manner. Job logs which are in the process of being written are allowed to continue until they have completed.

***IMMED**
The server is ended immediately. Jobs which are currently being processed by the job log server will not be removed from the system and the job logs currently being written will not be completed. The incomplete job logs will be rewritten from the beginning when a new job log server is started.

Top

Examples

```
ENDLOGSVR
```

This command ends the job log server in a controlled manner. Any job logs which are currently being written to spooled files by the job log server are allowed to complete.

Top

Error messages

***ESCAPE Messages**

CPF134B
Job log server not ended.

Top

End Mode (ENDMOD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Mode (ENDMOD) command ends (deactivates) a single mode or all active modes for a specific advanced program-to-program communications (APPC) remote location. The mode remains inactive until a Start Mode (STRMOD) command is run to start the mode. This command can be used to end all the sessions for a particular remote location and to cause an active switched connection to disconnect. The user can also specify how activities that have been requested on the remote system but have not yet been performed are to be handled.

The APPC Programming book, SC41-5443 has more information on the ENDMOD command.

Restriction: This command cannot be used to end (deactivate) Client Access/400 mode (QPCSUPP) at a remote location.

Top

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	*Communications name*	Required, Positional 1
DEV	Device	*Name*, ***LOC**	Optional
MODE	Mode	*Communications name*, ***NETATR**, *ALL	Optional
LCLLOCNAME	Local location	*Communications name*, ***LOC**, *NETATR	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NETATR, *NONE	Optional
CPLPNDRQS	Complete pended requests	***NO**, *YES	Optional

Top

Remote location (RMTLOCNAME)

Specifies the remote location name for which one or more modes are to be ended.

This is a required parameter.

Top

Device (DEV)

Specifies the device description name.

The possible values are:

***LOC** The device description is determined by the system.

device-name
Specify a device description name.

Top

Mode (MODE)

Specifies the mode that is to be ended.

The possible values are:

***NETATR**
The mode in the network attributes is used.

***ALL** All modes currently in use by the remote location are ended.

BLANK
The mode name (consisting of 8 blank characters) is used.

mode-name
Specify a mode name.

Note: SNASVCMG and CPSVCMG are reserved names and cannot be specified.

Top

Local location (LCLLOCNAME)

Specifies the local location name.

The possible values are:

***LOC** The local location name is determined by the system.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

local-location-name
Specify the name of your location. The local location name is specified if you want to indicate a specific local location name for the remote location.

Top

Remote network identifier (RMTNETID)

Specifies the remote network ID used with the remote location.

The possible values are:

***LOC** The system selects the remote network ID.

***NETATR**
The remote network identifier specified in the network attributes is used.

***NONE**
No remote network identifier (ID) is used.

remote-network-id
Specify the name of the remote network ID used.

Top

Complete pended requests (CPLPNDRQS)

Specifies if the remote location can complete pending work, or if the pended work should be ended before being allowed to start.

The possible values are:

_*NO_ The requested activities currently in progress at the remote location can complete. Activities that have been requested but not started at the remote location will not be performed.

***YES** All requested activities are allowed to complete before the mode is ended.

Top

Examples

```
ENDMOD   RMTLOCNAME(APPCRLOC)  MODE(APPCMOD)
```

This command ends a mode named APPCMOD for remote location APPCRLOC.

Top

Error messages

*ESCAPE Messages

CPF598B
The &1 command failed for one or more modes.

Top

End Mail Server Framework (ENDMSF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Mail Server Framework (ENDMSF) command ends the mail server framework jobs in the system work subsystem (QSYSWRK).

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 1
DELAY	Controlled end delay time	1-999999, **30**	Optional, Positional 2

Top

How to end (OPTION)

Specifies whether the mail server framework jobs that are in the system work subsystem (QSYSWRK) end immediately or in a controlled manner.

The possible values are:

***CNTRLD**
All mail server framework jobs are ended in a controlled manner. This allows each framework job a chance to complete processing the current mail server framework messages before it ends.

***IMMED**
All mail server framework jobs are ended immediately. Any mail server framework messages being processed at the time the job ended are processed when the mail server framework is restarted.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed for the mail server framework jobs to complete their processing during a controlled end. This parameter is ignored if OPTION(*IMMED) is specified. If the jobs do not end before the end of the delay time, they are then immediately ended.

The possible values are:

30 A maximum delay time of 30 seconds is allowed before the mail server framework jobs are ended.

delay-time

Specify the maximum amount of delay time in seconds before the jobs are ended. Valid values range from 1 through 999999.

Top

Examples

Example 1: Ending Mail Server Framework in a Controlled Manner

```
ENDMSF   OPTION(*CNTRLD)  DELAY(60)
```

This command ends the mail server framework jobs in the system work subsystem in a controlled manner and has 60 seconds to complete processing any mail server framework messages currently being handled.

Example 2: Ending Mail Server Framework Immediately

```
ENDMSF   OPTION(*IMMED)
```

This command ends the mail server framework jobs in the system work subsystem immediately. The mail server framework jobs do not complete processing any mail server framework messages currently being handled.

Top

Error messages

***ESCAPE Messages**

CPFAFAB

ENDMSF did not complete successfully.

CPFAFAC

ENDMSF completed successfully; however errors occurred.

CPFAFFF

Internal system error in program &1.

Top

End NFS Server (ENDNFSSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Network File System Server (ENDNFSSVR) command ends one or all of the NFS server daemons. For more information about these daemon jobs, see i5/OS Network File System Support book, SC41-5714.

SERVER(*ALL) should be specified, which will end the daemons in the following order. (This order is the recommended order for ending the Network File System daemons.)

- The network lock manager (NLM) daemon
- The network status monitor (NSM) daemon
- The mount (MNT) daemon
- The server (SVR) daemon(s)
- The name registry (RGY) daemon
- The generic security service (GSS) daemon
- The block input/output (I/O) (BIO) daemon(s)
- The Remote Procedure Call (RPC) RPCBind daemon

If just one daemon is to be ended, be sure the appropriate order for ending NFS daemons and the possible consequences of ending daemons in an order other than that specified above are understood. For more information about ending NFS daemons, see the i5/OS Network File System Support book, SC41-5714.

If the user attempts to end a daemon or daemons that are not running, they will not cause the command to fail, and it will continue to end other daemons that were requested to end.

To determine if an NFS daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following jobs:

```
QNFSRPCD  The RPCBind daemon
QNFSBIOD  The block I/O (BIO) daemon
QNFSGSSD  The generic security service (GSS) daemon
QNFSRGYD  The name registry (RGY) daemon
QNFSNFSD  The server (SVR) daemon
QNFSMNTD  The mount (MNT) daemon
QNFSNSMD  The network status monitor (NSM) daemon
QNFSNLMD  The network lock manager (NLM) daemon
```

Restrictions:

- The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
- The user must have job control (*JOBCTL) special authority to end any daemon jobs that were started by someone else.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server daemon	*ALL, *RPC, *BIO, *GSS, *RGY, *SVR, *MNT, *NSM, *NLM	Required, Positional 1
ENDJOBTIMO	Timeout for end of daemon	1-3600, **30**, *NOMAX	Optional, Positional 2

Top

Server daemon (SERVER)

Specifies the Network File System (NFS) daemon jobs to be ended.

***ALL** All NFS daemons will be ended.

***RPC** The NFS Remote Procedure Call (RPC) RPCBind daemon will be ended.

***BIO** All NFS block input/output (I/O) daemons that are running will be ended.

***GSS** The NFS generic security services daemon will be ended.

***RGY** The NFS name resolution registry daemon will be ended.

***SVR** All NFS server daemons that are running will be ended.

***MNT** The NFS mount daemon will be ended.

***NSM** The NFS network status monitor daemon will be ended.

***NLM** The NFS network lock manager daemon will be ended.

This is a required parameter.

Top

Timeout for end of daemon (ENDJOBTIMO)

Specifies the number of seconds to wait for each daemon to successfully end. If a daemon has not ended within the timeout value, the command will fail.

30 Wait 30 seconds for the daemon job to end.

***NOMAX**
Wait forever for daemons to end; do not timeout.

1-3600 Specify the number of seconds to wait for daemons to end before timing out and failing the command. Timeout values less than 30 seconds are rounded up to 30 seconds.

Top

Examples

Example 1: End All Daemons

```
ENDNFSSVR   SERVER(*ALL)
```

This command ends all NFS daemon jobs that are running.

Example 2: End a Single Daemon

```
ENDNFSSVR   SERVER(*MNT)  ENDJOBTIMO(*NOMAX)
```

This command ends the NFS mount daemon, and waits forever for it to end. The mount daemon was previously running, and other daemons have been ended in the appropriate order.

Top

Error messages

***ESCAPE Messages**

CPFA0B1
Requested operation not allowed. Access problem.

CPFA1B8
*IOSYSCFG authority required to use &1.

Top

End Network Interface Recovery (ENDNWIRCY)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Network Interface Recovery (ENDNWIRCY) command ends automatic error recovery procedures for a network interface description.

Top

Parameters

Keyword	Description	Choices	Notes
NWI	Network interface description	*Name*	Required, Positional 1

Top

Network interface description (NWI)

Specifies the name of the network interface description whose recovery is to be ended.

This is a required parameter.

Top

Examples

```
ENDNWIRCY   NWID(ISDNNET)
```

This command ends automatic error recovery procedures for the network interface named ISDNNET.

Top

Error messages

***ESCAPE Messages**

CPF591A
Not authorized to network interface description &1.

CPF593A
Network interface &1 not varied on.

CPF593B
Network interface description &1 not found.

CPF593C
Cannot access network interface &1.

Top

End Pass-Through (ENDPASTHR)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The End Pass-Through (ENDPASTHR) command ends a pass-through session. The ENDPASTHR command signs you off the target system, and ends the advanced program-to-program communications (APPC) session. This releases the virtual display device from the subsystem and returns it to the vary-on pending condition. The job at each intermediate node for the pass-through session also ends. Control returns to the source system for the next command following the Start Pass-Through (STRPASTHR) command.

Note: The ENDPASTHR command uses the SIGNOFF command as part of its processing. If the system has a SIGNOFF command that appears in the library list before QSYS/SIGNOFF, the SIGNOFF command is used by ENDPASTHR. The SIGNOFF command should not use the ENDPASTHR command. It sends the system into a loop when you end your pass-through session.

The ENDPASTHR command does not end the passthrough session when there is a secondary interactive job at the target system. One of the jobs must be ended (by using SIGNOFF or ENDJOB) before the ENDPASTHR command can be entered.

If the ENDPASTHR command is entered and there is not a pass-through session, an error message is sent.

More information about pass-through is in the Remote Work Station Support book, SC41-5402.

Top

Parameters

Keyword	Description	Choices	Notes
LOG	Job log	***NOLIST**, *LIST	Optional, Positional 1

Top

Job log (LOG)

Specifies whether the job log is saved at the target system.

***NOLIST**
The information in the job log is deleted when the job ends.

***LIST** A job log is saved at the target system.

Top

Examples

```
ENDPASTHR   LOG(*LIST)
```

This command ends a pass-through session and prints a job log.

Top

Error messages

***ESCAPE Messages**

CPF8914
ENDPASTHR command not allowed.

CPF8915
ENDPASTHR not allowed. System request job active.

Top

End Performance Explorer (ENDPEX)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End Performance Explorer (ENDPEX) command instructs the Performance Explorer tool to stop collecting data. The command expects a session name to accompany the request which identifies which instance of the Performance Explorer session to end.

The user can either end the data collection session or suspend the data collection session. If the user chooses to end the session, the collected data is put into an object of type *MGTCOL or into a set of database files, or it is deleted, based on the value specified for the DTAOPT parameter.

If the user chooses to suspend the collection of performance data, the session remains active. To resume data collection for a suspended session, the user can specify OPTION(*RESUME) on a subsequent call of the STRPEX (Start Performance Explorer) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have *ADD and *EXECUTE authority to the specified DTALIB and MGTCOL libraries.
3. The user must have *OBJMGMT, *OBJEXIST, and use authorities to the managment collection object if replacing an existing management collecton object.
4. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, 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.
5. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSRV
6. If running ENDPEX from a secondary thread, the QAYPE* files must already exist in the DTALIB library. These files must be created in the primary thread by running ENDPEX DTAOPT(*LIB).
7. If running ENDPEX from a secondary thread, DTAOPT(*MGTCOL) object cannot be specified.
8. Two threads within the same job will not be allowed to run ENDPEX at the same time. The thread that issued ENDPEX first will run the command to completion while the second ENDPEX waits.

Top

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	*Name*, ***SELECT**	Optional, Positional 1
OPTION	Option	***END**, *SUSPEND, *STOP, *EXTRACT	Optional
DTAOPT	Data option	***LIB**, *MGTCOL, *DLT	Optional
DTALIB	Data library	*Name*, **QPEXDATA**	Optional
DTAMBR	Data member	*Name*, ***SSNID**	Optional

Keyword	Description	Choices	Notes
MGTCOL	Management collection	*Qualified object name*	Optional
	Qualifier 1: Management collection	*Name*, ***SSNID**	
	Qualifier 2: Library	*Name*, **QPEXDATA**	
JOB	Jobs	Single values: ***ALL**, *NONE Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Job name	*Qualified job name*	
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000001-999999	
	Element 2: Thread identifier	Single values: ***ALL** Other values (up to 20 repetitions): 00000000-FFFFFFFF, *INITIAL	
TASK	Task name	Single values: ***ALL**, *NONE Other values (up to 8 repetitions): *Name*	Optional
RPLDTA	Replace data	***NO**, *YES	Optional
NBRTHD	Number of threads	1-256, ***CALC**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Session ID (SSNID)

Specifies which Performance Explorer session to end. This is the session identifier that was specified on the STRPEX (Start Performance Explorer) command.

***SELECT**
A list panel of all active Performance Explorer data collection sessions will be displayed with an option to select which session to end. *SELECT is only valid if the ENDPEX command is being run interactively. If the command is being run in batch, a session identifier must be specified.

session-identifier
Specify the Performance Explorer data collection session to end.

Top

Option (OPTION)

Specifies whether to end the data collection session or just suspend collection of performance data for the session.

***END** The Performance Explorer session is ended. The user can choose three methods to handle the collected data:

1. Save the collected data to a set of database files.
2. Save the data to a single file (used for sending data to IBM for analysis).
3. Discard the data.

***SUSPEND**
The Performance Explorer session is suspended, and the session remains active but no additional data is collected for this session. Once a session is suspended, the user can use STRPEX with OPTION(*RESUME) to resume data collection, end the suspended session by specifying ENDPEX with OPTION(*END), extract data with the *EXTRACT option, or stop the suspended session by specifying ENDPEX with OPTION(*STOP).

***STOP**
The Performance Explorer session is ended and the jobs are removed from the collection. The session cannot be started up again. Addresses are not resolved to object names, and no database files are created. The address data and database files can be created at a later time with the OPTION(*END) and DTAOPT(*LIB or *MGTCOL) options of ENDPEX. However, Performance Explorer may not be able to resolve some of the addresses if objects get deleted. The longer the time between *STOP and *END, the greater the chance the resolved address data will be incomplete.

***EXTRACT**
Data will be extracted from a suspended collection and saved to a set of database files. Before data is extracted, the session must be suspended with the ENDPEX OPTION(*SUSPEND) command. After data is extracted, the session can be resumed with the STRPEX OPTION(*RESUME) command or deleted with the ENDPEX OPTION(*END) DTAOPT(*DLT) command.

Top

Data option (DTAOPT)

Specifies how to handle the collected data. The collected data can be stored in a set of database files or a management collection object (*MGTCOL), or both. The temporary management collection object used to hold the collected data will be deleted. To delete the temporary management collection object without storing the collected data, specify *DLT.

Note: This parameter is valid only if OPTION(*END) is specified.

***LIB** Indicates to store all of the collected performance data for the session into a set of database files located in the library specified on the DTALIB parameter. The Performance Explorer tool creates all the necessary files if this is the first time that a library is being used to store performance data. The member name for each of the files where the session data is stored can be controlled through the DTAMBR parameter, but defaults to be the same name as the session identifier.

***MGTCOL**
Indicates to store all of the collected data in a management collection object (type *MGTCOL). No database files will be created. This option can be used if the data is to be shipped to another system or to your service provider for analysis.

***DLT** The collected performance data for the session is to be deleted from the system.

Top

Data library (DTALIB)

Specifies the name of the library that contains the set of database files where the collected performance data is stored.

Note: This parameter is valid only if the user specified DTAOPT(*LIB).

QPEXDATA
The QPEXDATA library is the recommended library for storing data collected with the Performance Explorer tool. The first time the Performance Explorer tool is used, this library is created for the user, and a set of database files to store the information is created in that library.

name Specifies the name of the library in which to store the collected data. If the library does not exist, the command ends in an error condition. After the library is created, retry the command. If the specified library does not already have the performance database files, they are created and the data is stored.

Top

Data member (DTAMBR)

Specifies the name to be used for the database file members where the collected performance data is stored. If a member does not exist by the specified name, it is created.

Note: This parameter is valid only when DTAOPT(*LIB) is specified.

***SSNID**
The member name is the same as the value specified for the SSNID parameter.

name Specify the member name to use when storing the collected data in Performance Explorer database files.

Top

Management collection (MGTCOL)

Specifies the management collection object where the the collected performance data is stored.

Note: This parameter is valid only if DTAOPT(*MGTCOL) is specified.

Qualifier 1: Management collection

***SSNID**
The name specified for the **Session ID (SSNID)** parameter is used when creating the management collection object to contain the collected performance data.

name Specify the name to use when creating the management collection object to contain the collected performance data.

Qualifier 2: Library

QPEXDATA
The QPEXDATA library is the recommended library for storing data collected by the Performance Explorer tool. The first time the Performance Explorer tool is used, this library is created for the user.

name Specify the name of the library to store the collected data. If the library does not exist, the command ends in an error condition. After the library is created, retry the command.

Top

Job name (JOB)

Specifies which job's data is saved from the Performance Explorer data collection session.

Single values

***ALL** Data is saved for all jobs in the collection.

***NONE**
No jobs have their data saved. Data is saved based on the TASK parameter.

Element 1: Job name

Qualifier 1: Job name

name Specify the name of the job to be included. Generic names are not allowed.

Qualifier 2: User

name Specify the user name of the jobs to be included. Generic names are not allowed.

Qualifier 3: Number

job-number
Specify the job number to further qualify the job name and user name.

Element 2: Thread identifier

***ALL** All threads of the specified job are included.

***INITIAL**
Only the initial thread of the specified job is included.

thread-identifier
Specify the thread identifier of the job to be included. This is the thread ID as shown by the WRKJOB command.

Top

Task name (TASK)

Specifies which licensed internal code (LIC) task's data is saved from the Performance Explorer data collection session. Up to 8 task names may be specified.

LIC tasks can be obtained from the Performance Tools reports and WRKSYSACT command. There is no guarantee that LIC task names will remain the same or exist from system to system or release to release.

Note: If the task name contains a hyphen, enclose the task name in quotation marks. For example, for task SK-ASC040, you would specify TASK("SK-ASC040").

Single values

***ALL** Data from all LIC tasks in the collection is included.

***NONE**
No data from LIC tasks in the collection is included. Data is includede based on the JOBS parameter.

Other values (up to 8 repetitions)

name Specify the name of the task to be included.

Top

Replace data (RPLDTA)

Specifies whether to replace the data in an existing file member or management collection object with the new performance data. If DTAMBR was specified and a member with the same name already exists in any of the Performance Explorer database files in the specified library (DTALIB parameter), this parameter controls whether the member data is replaced. If MGTCOL was specified and an object already exists with the same name, this parameter controls whether the data in that object is replaced.

***NO** If a member already exists with the same name, an error message is sent to the user. This prevents the user from inadvertently writing over existing data.

***YES** If a member already exists with the same name, the old data is lost and is replaced by the new data.

Top

Number of threads (NBRTHD)

Specifies the number of concurrent threads that the ENDPEX command uses to process the data in the session being ended. Specifying a number greater than 1 allows the ENDPEX command to take advantage of available CPU cycles, especially on a multi-processor system. While this may speed up the command processing, it may also degrade the performance of other jobs on the system. You can minimize this impact by changing the priority of the job that runs the ENDPEX command to a higher number. You should also verify that the disk subsystem can handle the additional threads.

***CALC**
The system calculates a reasonable number of threads to do the command processing which does not use excessive CPU or disk resources.

1-256 Specify the number of threads for the ENDPEX command to use to process the collected data.

Top

Text ’description’ (TEXT)

Specifies the text that briefly describes the type of data collected.

***BLANK**
Text is not specified.

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

Top

Examples

Example 1: End a Session and Save the Database Files

```
ENDPEX   SSNID(TEST3)  OPTION(*END)  DTAOPT(*LIB)
         DTAMBR(SYS1DATA)
```

This command ends the performance explorer session named TEST3 and saves the data in a set of database files in library QPEXDATA. The member name to be used for each file is SYS1DATA.

Example 2: End a Session and Delete the Data

```
ENDPEX   SSNID(TESTRUN)  OPTION(*END)  DTAOPT(*DLT)
```

This command ends the performance explorer session named TESTRUN and deletes the collected performance data.

Example 3: End a Session and Save the *MGTCOL

```
ENDPEX   SSNID(TEST3)  OPTION(*END)  DTAOPT(*MGTCOL)
         MGTCOL(MYLIB/SYS1DATA)  NBRTHD(*CALC)
```

This command ends the performance explorer session named TEST3 and saves the data in a management collection object in library MYLIB in the management collection object named SYS1DATA. ENDPEX will calculate a number of threads to process this request. This number of threads will do the ENDPEX processing as quickly as possible without disrupting the rest of the system.

Example 4: Suspend a Session and Extract the Data

```
ENDPEX   SSNID(TEST3)  OPTION(*SUSPEND)
```

```
ENDPEX   SSNID(TEST3)  OPTION(*EXTRACT)  DTAOPT(*LIB)
         DTAMBR(USERADATA) JOB(000123/USERA/QPADEV0001)
```

```
STRPEX   SSNID(TEST3)  OPTION(*RESUME)
```

This example first suspends the performance explorer session named TEST3. While the collection is suspended, the *EXTRACT option is used to save data collected for job 000123/USERA/QPADEV0001 to a collection named USERADATA. After the data is extracted, the collection is resumed and will begin collecting data again. This technique can be used when trying to find an intermittent problem.

Top

Error messages

*ESCAPE Messages

CPFAF06

ENDPEX command was not successful. Reason code is &1. See details for more information.

Top

End Performance Collection (ENDPFRCOL)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Performance Collection (ENDPFRCOL) command stops the system-level collection. If there are no other client applications using Collection Services, the Collection Services server job (QYPSPFRCOL) will also end.

If client applications are using Collection Services, the server job will continue to run unless you also specify the Force end parameter. Forcing the server job to end when it is being used by client applications (for example, IBM System i Navigator monitors or Performance Collector APIs) will cause those clients to experience data collection failure.

Other system functions are capable of starting Collection Services. Even though this command ends the current instance of the server job, it does not prevent the server job from being restarted. Functions which can start the server job include PM Agent, the Management Central server, and the Performance Collector APIs.

Top

Parameters

Keyword	Description	Choices	Notes
FRCCOLEND	Force end	***NO**, *YES	Optional, Positional 1

Top

Force end (FRCCOLEND)

Determines whether the Collection Services server job (QYPSPFRCOL) should be forced to end.

***NO** The QYPSPFRCOL job will be ended only if it is not being used by a client application.

***YES** The QYPSPFRCOL job will be forced to end immediately.

Top

Examples

Example 1: Ending the Performance Collection

```
ENDPFRCOL
```

This command will end the system-level collection of performance data. If no client applications are using Collection Services, this command will also end the Collection Services server job (QYPSPFRCOL). If client applications are using Collection Services, the QYPSPFRCOL job will continue to run.

Example 2: Forcing the Performance Collection to End

```
ENDPFRCOL FRCCOLEND(*YES)
```

This command will force the Collection Services server job (QYPSPFRCOL) to end, even if it is being used by client applications.

Top

Error messages

***ESCAPE Messages**

CPF3CF2
Error(s) occurred during running of &1 API.

CPFB94A
Collector communications error. Reason code &1.

Top

End Performance Trace (ENDPFRTRC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Performance Trace (ENDPFRTRC) command will stop the collection of performance trace data in the QPM_STRPFRTRC trace table and optionally write performance trace data to a data base file. The QPM_STRPFRTRC trace table will be deleted whether or not the data is written to a data base file.

This command is intended to be used to end a performance trace started via the Start Performance Trace (STRPFRTRC) command. However, it will end and try to process any active trace in the QPM_STRPFRTRC trace table.

Restrictions:

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

Top

Parameters

Keyword	Description	Choices	Notes
DMPTRC	Dump the trace	*NO, ***YES**	Optional, Positional 1
MBR	Member	*Name*	Optional
LIB	Library	*Name*, **QPFRDATA**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Dump the trace (DMPTRC)

Specifies whether the trace data is to be dumped to the performance database file QAPMDMPT. If the data is not dumped, it will be lost when the trace table is deleted.

***YES** The trace data, if any, is dumped.

***NO** The trace data is not dumped.

Top

Member (MBR)

Specifies the member within the QAPMDMPT database file where the trace table data is to be dumped. A value must be specified for this parameter if *YES is specified for the **Dump the trace (DMPTRC)** parameter.

name Specify the name of the database file member to be used.

Top

Library (LIB)

Specifies the library where the database file for trace data is located. If the file is not found in the specified library, the system automatically creates it in that library.

QPFRDATA
IBM-supplied performance data library QPFRDATA is to be used to locate the database file for trace data.

name Specify the name of the library to be searched.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the database member.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Ending Performance Trace

```
ENDPFRTRC   DMPTRC(*YES)  MBR(MYDATA)
```

In this example, the current trace is ended, the data is written to member MYDATA of file QAPMDMPT in library QPFRDATA, and the trace table is deleted, releasing the storage used by the trace.

Top

Error messages

***ESCAPE Messages**

Refer to the TRCINT and DMPTRC commands for messages.

Top

End Program (ENDPGM)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The End Program (ENDPGM) command specifies the end of a CL program or ILE CL procedure. When the command is processed, it performs the same function as a RETURN command. That is, control is returned to the command immediately following the CALL command in the calling program.

The ENDPGM command is not required at the end of a CL program or ILE CL procedure. If the last statement in a CL program or ILE CL procedure source file is reached and no ENDPGM command is found, an ENDPGM command is assumed by the compiler.

Restrictions: This command is valid only within a CL program or ILE CL procedure.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
PGM
  :
ENDPGM
```

This CL program or ILE CL procedure is identified by a PGM command that contains no parameters and is ended by the ENDPGM command.

Top

Error messages

None

Top

End Program Export List (ENDPGMEXP)

Parameters
Examples
Error messages

The End Program Export List (ENDPGMEXP) binder definition statement ends a list of exports in a service program export block.

There are no parameters for this statement.

Top

Parameters

None

Top

Examples

```
ENDPGMEXP
```

This binder definition statement marks the end of a list of exported variables or procedures for a service program.

Top

Error messages

None

Top

End Program Profiling (ENDPGMPRF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Program Profiling (ENDPGMPRF) command ends collection of program profiling data for programs or service programs that have been enabled to collect profiling data using the PRFDTA(*COL) option on the CHGPGM (Change Program), CHGSRVPGM (Change Service Program) CL command, or when the modules are created using the CHGMOD (Change Module) CL command.

Restrictions:

- This command is shipped with no public (*EXCLUDE) authority, and QPGMR user profile having use (*USE) authority to the command.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDPGMPRF
```

This command ends program profile data collection.

Top

Error messages

***ESCAPE Messages**

CPF5CAA
Unexpected error occurred during program profiling.

Top

End Prestart Jobs (ENDPJ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Prestart Jobs (ENDPJ) command ends all jobs and any associated inline data files for a prestart job entry in an active subsystem.

Jobs can be waiting for a request or can already be associated with a request. Spooled output files associated with the jobs being ended can also be ended or allowed to remain on the output queue. The limit on the number of messages being written to each of the job logs can also be changed.

Restrictions:

1. To use this command, you must have:
   - job control (*JOBCTL) special authority.
   - execute (*EXECUTE) authority to the library that contains the program.
   - use (*USE) authority to all auxiliary storage pool (ASP) device descriptions in the ASP group if the subsystem description specifies an ASP group.
2. Spooled output files on output queues in independent auxiliary storage pools (ASPs 33-255) are not deleted unless the subsystem description specifies an ASP group and the spooled output files are on output queues in that ASP group.

Top

Parameters

Keyword	Description	Choices	Notes
SBS	Subsystem	*Name*	Required, Positional 1
PGM	Program	*Qualified object name*	Required, Positional 2
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 3
DELAY	Controlled end delay time	1-999999, **30**	Optional, Positional 4
SPLFILE	Delete spooled files	***NO**, *YES	Optional, Positional 5
LOGLMT	Maximum log entries	*Integer*, ***SAME**, *NOMAX	Optional

Top

Subsystem (SBS)

Specifies the name of the active subsystem that contains the prestart job entry.

This is a required parameter.

name Specify the name of the active subsystem that contains the prestart job entry.

Top

Program (PGM)

Specifies the qualified name of the program that identifies the prestart job entry.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program that identifies the prestart job entry.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the program is located.

Top

How to end (OPTION)

Specifies whether the jobs are ended in a controlled manner, which lets the application program perform end-of-job processing, or the jobs are ended immediately.

***CNTRLD**
The jobs are ended in a controlled manner. This allows the program that is running to perform end-of-job processing. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. The application has the amount of time specified on the DELAY parameter to complete cleanup before the job is ended.

***IMMED**
The jobs end immediately. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job and the QENDJOBLMT system value specifies the time limit. Other than by handling the SIGTERM signal, the program that is running is not allowed to perform end-of-job processing.

Note: The *IMMED value might cause undesirable results if data has been partially updated. This value should be used only after a controlled end has been attempted unsuccessfully.

Top

Controlled end delay time (DELAY)

Specifies the time (in seconds) allowed for the program to complete end-of-job processing during a controlled end. If the end-of-job processing is not completed before the end of the delay time, the job is immediately ended. Only system cleanup is performed. This parameter is not valid if *IMMED is specified for the **How to end (OPTION)** parameter.

30 A maximum delay time of 30 seconds is allowed for end-of-job processing before each prestart job is ended.

1-999999
Specify the maximum delay time (in seconds) before each prestart job is ended.

Top

Delete spooled files (SPLFILE)

Specifies whether spooled output files created by the jobs are retained for normal processing by a writer or deleted.

***NO** The spooled output files created by the jobs being ended are retained for normal processing by a writer. When the job ends, the spooled file action (SPLFACN) job attribute determines whether spooled files are detached from the job or kept with the job.

***YES** The spooled output files created by the jobs being ended and which are on output queues in the system auxiliary storage pool (ASP 1) or in a basic user ASP (ASPs 2-32) are deleted. Spooled output files on output queues in independent ASPs (ASPs 33-255) are not deleted. The job log is not deleted.

Top

Maximum log entries (LOGLMT)

Specifies the maximum number of entries in the message queue of the jobs being ended that are written to the job log. This parameter can be used to limit the number of messages written to the job log printer file, QPJOBLOG, for each job that is ended.

***SAME**
The message logging limit does not change. If the logging limit was not changed for these prestart jobs on a previous command, *NOMAX is the value used by the system.

***NOMAX**
There is no limit to the number of messages being logged. All messages on each job message queue are written to the job log for each job.

integer-number
Specify the maximum number of messages being written to the job log for each job. This value is the maximum only if it is entered before the job log contains that number of messages. Otherwise, the limit only stops the process of writing any more messages to the job log. If 0 is specified before any messages are written to the log, no job log is produced.

Top

Examples

Example 1: Ending a Job Immediately

```
ENDPJ   SBS(SBS1)  PGM(PJLIB/PJPGM)  OPTION(*IMMED)
        SPLFILE(*YES)
```

This command ends all jobs associated with prestart job entry PJPGM in subsystem SBS1 immediately. Spooled output produced by these prestart jobs is deleted and the job log is saved.

Example 2: Delaying a Job End

```
ENDPJ   SBS(SBS2)  PGM(PJPGM2)  OPTION(*CNTRLD)
        DELAY(50)  SPLFILE(*NO)
```

This command ends all the jobs associated with prestart job entry PJPGM2 in subsystem SBS2. Spooled output for these prestart jobs is saved for normal processing by the spooling writer. The jobs have 50 seconds to perform any cleanup routines, after which they are immediately ended.

Top

Error messages

***ESCAPE Messages**

CPF0922
End Prestart Jobs command not allowed now.

CPF101C
Not authorized to device &1.

CPF1083
Prestart jobs already are ending controlled.

CPF1084
Prestart jobs are already ending immediately.

CPF1227
No authority has been granted to use command.

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1834
Prestart job entry for program &1 in &2 does not exist.

CPF9820
Not authorized to use library &1.

Top

End Printer Emulation (ENDPRTEML)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Printer Emulation (ENDPRTEML) command ends printer emulation without ending the job. If there is another request in the job, that request is then processed.

This command closes the file to the host system, then writes the last data received from the host system out to the spooled file or printer by closing the printer file.

In some cases, the request does not take effect immediately. The request is delayed while any of the following conditions exist in the printer emulation request:

- Printing a block sent from the host system.
- Waiting for a printer error to be cleared (for example, a paper jam).
- Waiting for a reply to a PA1 or PA2 inquiry message.
- Waiting for error recovery to be done to the host system or printer device.
- The job has been held by using the HLDJOB command. When the condition is cleared, the End Printer Emulation request takes effect, and the printer emulation request ends.

Top

Parameters

Keyword	Description	Choices	Notes
EMLDEV	Emulation device, or	*Name*	Optional, Positional 1
EMLLOC	Emulation location	*Communications name*	Optional, Positional 2
PRTDEV	Print device	*Name*	Optional, Positional 3

Top

Emulation device (EMLDEV)

Specifies the name of a printer emulation device that receives data from the host system. This device must be a 3287 Printer (EMLDEV(3287)) or a 3289 Printer (EMLDEV(3289)), and must currently be operating as an LU1 unit. The printer emulation job or session that is using this device will be informed of the request. If the LU1 session is between brackets, printer emulation starts a bracket and sends the PA key signal to the host system with a Change Direction (CD) request. If the LU session is in receive condition, a signal (request for CD) is sent to the host system, and printer emulation waits for the CD. When the CD is received, the PA key signal is sent to the host system with the CD. If the LU session is in send condition, the PA key signal is sent to the host system with the CD.

Either this parameter, or the **Emulation location (EMLLOC)** parameter and the **Print device (PRTDEV)** parameter is required.

Top

Emulation location (EMLLOC)

Specifies the remote location name associated with this session. The location name is defined during device description configuration and refers to the remote location where communication takes place. This value must be the same as the value specified for the Emulation location (EMLLOC) parameter on the Start Printer Emulation (STRPRTEML) command.

Either this parameter and the **Print device (PRTDEV)** parameter, or the **Emulation device (EMLDEV)** parameter is required.

Top

Print device (PRTDEV)

Specifies the name of a printer device that is used to print the spooled output. This value must be the same as the value specified for the Printer device (PRTDEV) parameter on the Start Printer Emulation (STRPRTEML) command. This parameter must be specified when the EMLLOC parameter is specified.

Either this parameter and the **Emulation location (EMLLOC)** parameter, or the **Emulation device (EMLDEV)** parameter is required.

Top

Examples

```
ENDPRTEML   EMLDEV(HOSTPRT3)
```

This command ends the printer emulation request that is using the device HOSTPRT3.

Top

Error messages

***ESCAPE Messages**

CPF8599
End printer emulation function not performed.

Top

End Receive (ENDRCV)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The End Receive (ENDRCV) command is used to end (cancel) a request for input made by a previously issued RCVF or SNDRCVF command that had WAIT(*NO) specified. The ENDRCV command ends an input request even if the user enters the requested data at the display station at the same time that the command is processed. If the requested data is entered and is being sent to the program when the end receive operation is performed, the entered data is lost. If there is no outstanding input request, the command is ignored.

Restrictions:

- This command is valid only for display files within a CL program or ILE CL procedure. It cannot be used for database files.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Display device	*Name*, ***FILE**	Optional, Positional 1
OPNID	Open file identifier	*Simple name*, ***NONE**	Optional

Top

Display device (DEV)

Specifies the name of the display device for which the request for input is to be ended.

***FILE** The name of the device whose response is to be ended. This device is in the device file that is declared in the **File (FILE)** parameter of the Declare File (DCLF) command. If the device file has more than one device name specified in it, *FILE cannot be specified.

name Specify the name of the display device from which a response is to be ended.

Top

Open file identifier (OPNID)

Specifies the open file identifier that was declared on a preceding Declare File (DCLF) command in the same CL program or ILE CL procedure. A CL variable cannot be specified for this parameter value.

***NONE**

No open file identifier is provided. This command will use the file associated with the DCLF command that had *NONE specified for the OPNID parameter. Only one file can be declared in a CL program or ILE CL procedure with *NONE as the open file identifier.

simple-name

Specify a name that matches the OPNID parameter value on a preceding DCLF command in the same CL program or ILE CL procedure.

Top

Examples

Example 1: Ending Previous Receive

```
ENDRCV   DEV(MYDISPLAY)
```

Assume that a RCVF command with WAIT(*NO) was issued earlier in the CL program or ILE CL procedure to request input from the device file declared earlier in the DCLF command and from the display device MYDISPLAY. When this ENDRCV command is processed, that request for input from MYDISPLAY is ended.

Example 2: Using an Open File Identifier

```
DCLF   FILE(MYLIB/MYDSPFILE)  RCDFMT(FMT1)  OPNID(DSPFILE1)
  :
SNDRCVF   DEV(DSP02)  RCDFMT(FMT1)  OPNID(DSPFILE1)  WAIT(*YES)
  :
ENDRCV   DEV(DSP02)  OPNID(DSPFILE1)
```

This command ends the previous SNDRCVF (Send/Receive File) command's request for input from a workstation display device DSP02.

Top

Error messages

***ESCAPE Messages**

CPF0883

*FILE not valid in DEV parameter for file &1.

CPF4101

File &2 in library &3 not found or inline data file missing.

Top

End Reader (ENDRDR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Reader (ENDRDR) command ends the specified database reader. The reader can be stopped either immediately, without completing the current job being read, or at the end of the current job. If the reader is in a hold state when this command is issued, the reader is stopped immediately.

Top

Parameters

Keyword	Description	Choices	Notes
RDR	Reader	*Name*	Required, Positional 1
OPTION	When to end reader	***CNTRLD**, *IMMED	Optional, Positional 2

Top

Reader (RDR)

Specifies the database reader to be ended.

This is a required parameter.

name Specify the name of the reader to be ended.

Top

When to end reader (OPTION)

Specifies when the ended reader should stop processing.

***CNTRLD**
The reader stops processing after the current job is read and an entry for the job is placed on the job queue.

***IMMED**
The reader stops processing immediately. The job being read is not placed on the job queue.

Top

Examples

```
ENDRDR   RDR(DISKETTE)
```

This command stops the reader DISKETTE as soon as the current job is completely read in and releases that device to the system.

Top

Error messages

***ESCAPE Messages**

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1352
Function not done. &3/&2/&1 in transition condition.

CPF3312
Reader &1 neither active nor on job queue.

CPF3330
Necessary resource not available.

CPF3490
Not authorized to specified reader.

Top

End Remote Support (ENDRMTSPT)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The End Remote Support (ENDRMTSPT) command varies off and deletes the line, controller's and device descriptions created by the Start Remote Support (STRRMTSPT) command. This command optionally deletes the QTILIB library created by the (STRRMTSPT) command.

Restriction: This command is not valid when you are signed-on the remote support work station.

Top

Parameters

Keyword	Description	Choices	Notes
DLTLIB	Delete library	*YES, ***NO**	Optional, Positional 1
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 2

Top

Delete library (DLTLIB)

Specifies if the remote service library (QTILIB) should be deleted when running the (ENDRMTSPT) command.

NO The remote service library (QTILIB) is not deleted.

YES Delete the remote service library (QTILIB).

Top

How to end (OPTION)

Specifies how the remote support connection is ended.

CNTRLD
The remote support connection ends when the connection timeout is reached.

IMMED
The remote support connection ends immediately.

Top

Examples

```
ENDRMTSPT   DLTLIB(*NO)  OPTION(*IMMED)
```

This command immediately ends the remote support connection and deletes the configuration objects that have been created.

Top

Error messages

None

Top

End RPC Binder Daemon (ENDRPCBIND)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End RPC Binder Daemon (ENDRPCBIND) command ends the Remote Procedure Call (RPC) RPCBind daemon. The RPC binder daemon job must be running to use and run Network File System (NFS) daemons and commands and some of the TI-RPC APIs.

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

- ENDNFSSVR SERVER(*RPC)

If the user attempts to end this daemon and it is not running, it will not cause the command to fail.

To determine if the RPCBind daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following job:

```
QNFSRPCD  The RPCBind daemon
```

Restrictions

- The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.

Top

Parameters

None

Top

Examples

Example 1: Ending the RPC Binder Daemon

```
ENDRPCBIND
```

This command ends the RPC binder daemon job, if it is running.

Top

Error messages

***ESCAPE Messages**

CPFA1B8
*IOSYSCFG authority required to use &1.

Top

End Request (ENDRQS)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The End Request (ENDRQS) command ends (cancels) a previously requested operation (command). One common use of the End Request (ENDRQS) command is to cancel a request that is currently stopped at a breakpoint. This command function is also available as an option on the System Request menu.

If the End Request (ENDRQS) command cannot be processed immediately because a system function that cannot be interrupted is currently running, the command is delayed until interruption is allowed.

When a request is ended, an escape message is sent to the request processing program that is currently called at the request level being canceled. Request processing programs can be canceled. Request processing programs can monitor for the escape message so that cleanup processing can be done when the request is canceled. The static storage and open files are reclaimed for any program that was called by the request processing program. None of the programs called by the request processing program is notified of the cancel, so they have no opportunity to stop processing.

To become a request processing program, the program must receive a request message.

If the ENDRQS command is in a program, that program must become a request processor before it issues this command.

More information on how to set up a program to become a request processor is in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: External objects that are locked by the Allocate Object (ALCOBJ) command are not unlocked (deallocated) by the canceled request.

Top

Parameters

Keyword	Description	Choices	Notes
RQSLVL	Request level	*Integer*, ***PRV**	Optional, Positional 1

Top

Request level (RQSLVL)

Specifies the (command) request level at which the command being canceled was entered.

***PRV** The command entered at the immediately previous level is being canceled.

integer
Specify the request level at which the command being canceled was entered. All request levels from the level specified to the current level are canceled.

Top

Examples

Example 1: Ending a Command

```
CALL PROGA     (This is level 1)
  :
Breakpoint occurs
CALL PROGB     (This is level 2)
  :
Breakpoint occurs
ENDRQS         (This is level 3)
```

In this example, because RQSLVL(*PRV) is the default, the request made at level 2 is canceled. The user can then enter another command at level 2 or press F3 to show the PROGA breakpoint display again.

Example 2: Ending a Command

```
CALL PROGA          (This is level 1)
  :
Breakpoint occurs
CALL PROGB          (This is level 2)
  :
Breakpoint occurs
ENDRQS  RQSLVL(1)   (This is level 3)
```

In this example, the request made at the highest level (CALL PROGA) is canceled. Consequently, any requests made between level 1 and level 3 are also canceled.

Top

Error messages

None

Top

End S/36 Session (ENDS36)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The End System/36 (ENDS36) command allows the user to end the System/36 environment session that was started with a Start System/36 (STRS36) command.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDS36
```

This command immediately ends the System/36 Environment session and any programs or procedures that are running in the System/36 Environment. If the ENDS36 command is in a procedure or in a program, the statements following the command are ignored.

Top

Error messages

None

Top

End Subsystem (ENDSBS)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Subsystem (ENDSBS) command ends the specified subsystem (or all active subsystems) and specifies what happens to active work being processed. No new jobs or routing steps are started in the subsystem or subsystems after this command is run.

Interactive jobs that have been transferred to a job queue by the Transfer Job (TFRJOB) command are ended as part of ending the subsystem. If an initial program load (IPL) occurs while either a batch or interactive job is on a job queue (because of the TFRJOB command), that job is removed from the job queue during IPL and its job log is produced.

You can specify that the application programs running in the subsystem are given time to control end-of-job processing. If no time is given or if cleanup cannot be performed within the given time, the system performs minimal end-of-job processing, which can include:

- Closing the database files.
- Spooling the job log to an output queue.
- Cleaning up internal objects in the operating system.
- Showing the end-of-job display (for interactive jobs).
- Completing commitment control processing.

Restrictions:

1. To use this command, you must have job control (*JOBCTL) special authority.
2. If the controlling subsystem is being ended, because either its name or *ALL is specified for the **Subsystem** (SBS) parameter, then this command can be run only in
   - an interactive job that is in the controlling subsystem and only from a work station (associated with the interactive job) whose work station entry in the controlling subsystem description specifies *SIGNON for the **Allocation** (AT) parameter. For more information, see the Add Work Station Entry (ADDWSE) command.
   - or a batch job running in the controlling subsystem, initiated from a job queue, with the BCHTIMLMT parameter and SBS(*ALL) specified.

   ENDSBS SBS(*ALL) is not allowed in a TELNET job, pass-through job, or in a workstation function job.
3. ENDSBS SBS(*ALL) is not allowed in a batch job that allows multiple threads.

Top

Parameters

Keyword	Description	Choices	Notes
SBS	Subsystem	*Name*, *ALL	Required, Positional 1
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 2
DELAY	Controlled end delay time	0-99999, ***NOLIMIT**	Optional

Keyword	Description	Choices	Notes
ENDSBSOPT	End subsystem option	Single values: ***DFT** Other values (up to 3 repetitions): *NOJOBLOG, *CHGPTY, *CHGTSL	Optional
BCHTIMLMT	Batch time limit	5-9999, *NOMAX	Optional

Top

Subsystem (SBS)

Specifies the name of the subsystem to be ended, or it specifies that all active subsystems are to be ended.

This is a required parameter.

***ALL** All the subsystems that are currently active are being ended. All jobs are ended except the job in which this command is entered. When this value is specified, the QSYSOPR message queue should be in break delivery mode in the job issuing the end subsystem command.

name Specify the simple name of the subsystem to be ended. If the subsystem specified is the controlling subsystem, the interactive job from which the command was issued remains active. Also, if the subsystem specified is the controlling subsystem and the job that issues this command is one of two secondary jobs that are active at the work station, neither of the jobs is forced to end. The controlling subsystem does not end until you end one of the jobs (either by signing off in one job or by ending one job from the other).

Top

How to end (OPTION)

Specifies whether jobs in the subsystem are ended in a controlled manner (ending jobs in a controlled manner lets the application programs perform end-of-job processing) or immediately.

***CNTRLD**

The jobs are ended in a controlled manner. This allows the programs that are running to perform cleanup (end of job processing). When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. The application has the amount of time specified for the DELAY parameter to complete cleanup before the job is ended.

***IMMED**

The jobs are ended immediately. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job and the QENDJOBLMT system value specifies a time limit. Other than by handling the SIGTERM signal, the programs that are running are not allowed to perform any cleanup.

Note: The *IMMED value might cause undesirable results if data has been partially updated. This value should be used only after a controlled end has been attempted unsuccessfully.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) that is allowed to complete the controlled subsystem end operation. If this amount of time is exceeded and the end operation is not complete, any jobs still being processed in the subsystem are ended immediately.

***NOLIMIT**
The amount of time in which to complete a controlled end is not limited.

0-99999
Specify the number of seconds in which the end operation is allowed to complete.

Top

End subsystem option (ENDSBSOPT)

Specifies the options to take when ending the active subsystems. In general, specifying these options will improve the performance of the ENDSBS command. Each option has certain side effects that you need to analyze before using that option.

This parameter has no effect on jobs that are already in the ending status.

***DFT** The subsystems will end with no special ending options.

- Joblogs will be produced.
- The run priority will not change.
- The timeslice value will not change.

***NOJOBLOG**
No joblogs will be created for jobs that are ended due to this command being invoked. This includes subsystem monitor jobs and all user jobs in the subsystem. This option can significantly reduce the amount of time necessary to complete the ENDSBS command. However, if a problem occurs in a job, there will be no joblog to record the problem, which may make problem diagnosis difficult or impossible.

***CHGPTY**
The CPU priority of jobs that are ending is changed to a higher value (worse priority). The remaining active jobs on the system may have better performance when *CHGPTY is specified. However, jobs that are ending may take longer to finish. This option is ignored if the subsystem is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

***CHGTSL**
The timeslice of jobs that are ending is changed to a lower value. The remaining active jobs on the system may have better performance when *CHGTSL is specified. However, jobs that are ending may take longer to finish. This option is ignored if the subsystem is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

Note: Specifying *CHGPTY and *CHGTSL will reduce the impact on other active jobs on the system, but this may cause undesirable delays if there are active workstations that were allocated to the ending subsystem. It may take longer for those workstations to have their signon screens re-displayed since the job using the display must end before the workstation is ready to be allocated to another subsystem.

Top

Batch time limit (BCHTIMLMT)

Specifies the amount of time (in minutes) that the system will run in batch restricted state. This parameter is only valid when ending all subsystems from a batch job running in the controlling subsystem. Under this condition, a parameter value must be specified. When this parameter is specified, the system will be ended to the restricted state, with only the batch job running the ENDSBS command remaining active. While the system is in this restricted state, system reference code A900 3C70 is displayed. If the specified time limit is reached, the batch job will be ended and the controlling subsystem restarted.

Note: This parameter is recommended only for an application that requires no operator interaction.

***NOMAX**
There is no time limit for the batch restricted function. The system will remain in the restricted state until the job ends, the Start Subsystem (STRSBS) command is used, or the Dedicated Service Tools (DST) option to end batch restricted state is used.

5-9999 Specify the time limit (in minutes) that the batch restricted function is allowed to run.

Top

Examples

```
ENDSBS   SBS(QBATCH)  OPTION(*CNTRLD)  DELAY(60)
```

This command ends all active jobs in the QBATCH subsystem and ends the subsystem. The active jobs are allowed 60 seconds to perform application-provided end-of-job processing.

Top

Error messages

***ESCAPE Messages**

CPF1001
Wait time expired for system response.

CPF1032
System ending with *CNTRLD option.

CPF1033
System ending with *IMMED option.

CPF1034
All subsystems ending with *CNTRLD option.

CPF1035
Subsystems ending with *IMMED option.

CPF1036
System powering down with *CNTRLD option.

CPF1037
System powering down with *IMMED option.

CPF1038
No authority to use command.

CPF1052
ENDSBS *ALL not allowed in current environment.

CPF1053
Ending controlling subsystem &1 not allowed.

CPF1054
No subsystem &1 active.

CPF1055
Subsystem &1 ending with *CNTRLD option.

CPF1056
Subsystem &1 already ending with *IMMED option.

CPF1081
Controlling subsystem already ending to a single job.

CPF1091
Function check occurred in system arbiter.

CPF18C3
Exit Point Program &1 cannot enter restricted state.

Top

End Select (ENDSELECT)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The End Select (ENDSELECT) command is used with the SELECT command to select a group of commands that are to be processed. The ENDSELECT command specifies the end of the select group that is started with an associated SELECT command. The ENDSELECT command must be specified after the last WHEN or or OTHERWISE command in the select group.

When select groups are nested, each group must have its own ENDSELECT command at its end. Every ENDSELECT command must be associated with a SELECT command; if too many ENDSELECT commands occur in the CL program or ILE CL procedure source, a message is issued and the program is not created.

Restrictions:

- This command is valid only within a CL program or ILE CL procedure.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
DCL   VAR(&NAME)  TYPE(*CHAR)  LEN(10)
 :
SELECT
  WHEN   COND(&NAME *EQ *CMD)  THEN(DO)
    :   (group of CL commands)
  ENDDO
  :   (other WHEN or OTHERWISE commands)
ENDSELECT
```

The ENDSELECT command ends an active SELECT command group.

Top

Error messages

None

Top

End Service Agent (ENDSRVAGT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Service Agent (ENDSRVAGT) command allows a user to end an aspect of Service Agent. The aspect to be ended is specified by the **Type (TYPE)** parameter.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type	***SBSJOB**	Optional, Positional 1

Top

Type (TYPE)

Specifies the aspect of Service Agent to be ended.

***SBSJOB**

All Service Agent monitoring jobs running in the QSYSWRK subsystem are to be ended immediately. This option will have no effect if the QSYSWRK subsystem has not been started or the Service Agent monitoring jobs have not been started in the QSYSWRK subsystem.

Top

Examples

```
ENDSRVAGT   TYPE(*SBSJOB)
```

This command will end immediately all Service Agent monitoring jobs running in the QSYSWRK subsystem.

Top

Error messages

***ESCAPE Messages**

CPF9899

Error occurred during processing of command.

Top

End Service Job (ENDSRVJOB)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Service Job (ENDSRVJOB) command ends the remote job service operation. This command stops the service operation that began when the Start Service Job (STRSRVJOB) command was entered.

Restrictions:

- If tracing or debugging is active in the serviced job when this command is entered, the remote service operation is *not* ended.
- The following user profiles have private authorities to use the command:
  - QPGMR
  - QSYSOPR
  - QSRV
  - QSRVBAS

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDSRVJOB
```

This command stops the service operation of the job currently being serviced.

Top

Error messages

None

Top

End Subroutine (ENDSUBR)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The End Subroutine (ENDSUBR) command specifies the end of a subroutine, and must be paired with a previous Subroutine (SUBR) command. When the ENDSUBR command is processed, control is returned to the command immediately following the Call Subroutine (CALLSUBR) command that called the subroutine. If a value is to be returned, such as an error code, the returned value must be able to be stored into a 4-byte signed integer CL variable. The **Return value (RTNVAL)** parameter can be a variable or constant. If no RTNVAL parameter is defined, the value will default to zero.

Restrictions:

- This command is valid only within a CL program or ILE CL procedure.
- The subroutine must be defined within a procedure, and must be located at the end of the procedure.

Top

Parameters

Keyword	Description	Choices	Notes
RTNVAL	Return value	*Integer*, **0**	Optional

Top

Return value (RTNVAL)

Specifies the value to be returned from the subroutine. This can be a variable or constant that can be stored in a 4-byte signed integer CL variable. If the Call Subroutine (CALLSUBR) command that called the subroutine specified a **CL variable for returned value**, the variable is set to this value.

0 The subroutine returns a zero.

integer-constant
Specify the constant integer value for the return value.

CL-variable-name
Specify the name of the CL variable to contain the return value from the called subroutine. This must be a signed integer CL variable.

Top

Examples

```
PGM
 :
SUBR SUBR1
 :
ENDSUBR
ENDPGM
```

This CL subroutine is identified by a SUBR command that contains the subroutine name, SUBR1, and is ended by the ENDSUBR command.

```
PGM
 :
SUBR SUBR1
 :
ENDSUBR RTNVAL(-1)
ENDPGM
```

This CL subroutine has an ENDSUBR command that returns a -1.

Top

Error messages

None

Top

End System (ENDSYS)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The End System (ENDSYS) command ends most activity on the system and leaves the system in a condition in which only the console is active in the controlling subsystem. This is done so that the operator can do things like backing up the system or loading new programs. This condition is called the restricted state and is required for operations like saving the system or reclaiming storage. If two jobs are active in the controlling subsystem at the console, neither of the jobs is forced to end. The ENDSYS command cannot complete running until you end one of the jobs (either by signing off in one job or by ending one job from the other).

All active subsystems are notified that an end system operation is in process. No new jobs or routing steps can be accepted by the subsystems. This command also specifies what happens to all active work.

Interactive jobs that are transferred to a job queue by the Transfer Job (TFRJOB) command are ended as part of subsystem ending. If an initial program load (IPL) occurs while either a batch or interactive job is on a job queue (because of the TFRJOB command), that job is removed from the job queue during IPL and its job log is produced.

Restriction: This command can be entered only in an interactive job in the controlling subsystem. To use this command, the user must have job control (*JOBCTL) authority. This command is not allowed in a pass-through job or in a workstation function job.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	***CNTRLD**, *IMMED	Optional, Positional 1
DELAY	Controlled end delay time	0-99999, ***NOLIMIT**	Optional
ENDSBSOPT	End subsystem option	Single values: ***DFT** Other values (up to 3 repetitions): *NOJOBLOG, *CHGPTY, *CHGTSL	Optional
CONFIRM	Confirm	***ENVVAR**, *YES, *NO	Optional

Top

How to end (OPTION)

Specifies whether all active jobs are ended in a controlled manner (which lets the application programs perform end of processing) or immediately. In either case, the system performs certain job cleanup functions.

***CNTRLD**
The jobs are ended in a controlled manner. This allows the programs that are running to perform cleanup (end of job processing). When a job being ended has a signal handling procedure for the

asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. The application has the amount of time specified for the DELAY parameter to complete cleanup before the job is ended.

***IMMED**
The jobs are ended immediately. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job and the QENDJOBLMT system value specifies a time limit. Other than by handling the SIGTERM signal, the programs that are running are not allowed to perform any cleanup.

Note: The *IMMED value might cause undesirable results if data has been partially updated. This value should be used only after a controlled end has been attempted unsuccessfully.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) that the controlled end operation is allowed. If this amount of time is exceeded and the end operation is not complete, any jobs still being processed are ended immediately, except for those running long-running instructions.

***NOLIMIT**
The amount of time in which to complete a controlled end is not limited.

0-99999
Specify the number of seconds in which the end operation is allowed to complete.

Top

End subsystem option (ENDSBSOPT)

Specifies the options to take when ending the active subsystems. In general, specifying these options will improve the performance of the ENDSYS command. Each option has certain side effects that you need to analyze before using that option.

This parameter has no effect on jobs that are already in the ending status.

***DFT** The subsystems will end with no special ending options.

- Joblogs will be produced.
- The run priority will not change.
- The timeslice value will not change.

***NOJOBLOG**
No joblogs will be created for jobs that are ended due to this command being invoked. This includes subsystem monitor jobs and all user jobs in the subsystem. This option can significantly reduce the amount of time necessary to complete the ENDSYS command. However, if a problem occurs in a job, there will be no joblog to record the problem, which may make problem diagnosis difficult or impossible.

***CHGPTY**
The CPU priority of jobs that are ending is changed to a higher value (worse priority). The remaining active jobs on the system may have better performance when *CHGPTY is specified. However, jobs that are ending may take longer to finish. This option is ignored if the subsystem is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

***CHGTSL**
The timeslice of jobs that are ending is changed to a lower value. The remaining active jobs on the system may have better performance when *CHGTSL is specified. However, jobs that are

ending may take longer to finish. This option is ignored if the subsystem is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

Top

Confirm (CONFIRM)

Specifies whether the request should be confirmed before the system is ended.

***ENVVAR**
The value in environment variable QIBM_ENDSYS_CONFIRM is used to determine whether the request should be confirmed. If the value is set to *YES or *NO, the action described below for that value is taken. If the environment variable is not defined or not set to one of these values, then there is no confirmation.

***YES** A confirmation panel is displayed when the ENDSYS command is issued.

***NO** There is no confirmation when the ENDSYS command is issued.

Top

Examples

Example 1: Ending System Activity

```
ENDSYS
```

This command ends the system activity after all active jobs in the system are allowed to perform their own end of processes. The amount of time the end can take is not limited.

Example 2: Ending System Activity After Jobs are Ended

```
ENDSYS   OPTION(*IMMED)
```

This command ends system activity after all active jobs are immediately ended.

Top

Error messages

***ESCAPE Messages**

CPF1001
Wait time expired for system response.

CPF1017
ENDSYS not allowed when console powered or varied off.

CPF1032
System ending with *CNTRLD option.

CPF1033
System ending with *IMMED option.

CPF1034
All subsystems ending with *CNTRLD option.

CPF1035
Subsystems ending with *IMMED option.

CPF1036
System powering down with *CNTRLD option.

CPF1037
System powering down with *IMMED option.

CPF1038
No authority to use command.

CPF1051
Command can only be run in controlling subsystem.

CPF1082
Controlling subsystem already ending to single job.

CPF1091
Function check occurred in system arbiter.

CPF18C3
Exit Point Program &1 cannot enter restricted state.

Top

End TCP/IP (ENDTCP)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End TCP/IP (ENDTCP) command ends TCP/IP processing.

Attention:

The ENDTCP command must be used carefully. When it is used, it ends all TCP/IP processing on the system that you are working on.

If OPTION(*IMMED) is specified for the ENDTCP command, the following is true:

- All TCP/IP connections are ended. This affects all currently active applications using sockets.
- Unless ENDSVR(*NO) is specified, TCP/IP server jobs are ended for TELNET, FTP, TFTP, SMTP, LPD, HTTP, POP, RouteD, DHCP, DNS, DDM, BOOTP, REXEC, SNMP, DIRSRV, NSLD, INETD, MGTC, ONDMD, NETSVR, DLFM, VPN, EDRSQL, HOD, ODPA, NTP, QoS, TCM, DOMINO, WEBFACING, OMPROUTED and CIMOM.
- Agents that are currently active in the QSYSWRK subsystem are ended. See the description of the **End application servers (ENDSVR)** parameter for more information.
- All active TCP/IP interfaces are ended.

If OPTION(*CNTRLD) is specified for the ENDTCP command, the following is true:

- No new open operations are allowed to TCP, UDP, or raw sockets.
- A job is submitted to the QSYSWRK subsystem that will, after the time indicated in the DELAY parameter value has expired, do an ENDTCP *IMMED operation.
- An ENDTCP OPTION(*IMMED) can be submitted at any time after issuing ENDTCP OPTION(*CNTRLD). This cancels the controlled end. TCP/IP processing is ended immediately when the ENDTCP OPTION(*IMMED) is issued.

Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	***IMMED**, *CNTRLD	Optional, Positional 1
DELAY	Controlled end delay time	1-86400, **30**	Optional, Positional 2
ENDSVR	End application servers	***YES**, *NO	Optional

Top

How to end (OPTION)

Specifies whether TCP/IP processing is ended in an immediate or controlled manner.

***IMMED**
TCP/IP processing is ended immediately.

Attention:

The ENDTCP OPTION(*IMMED) command should be used carefully. Partially updated data may result if an application is processing data and has not completed an operation when the ENDTCP *IMMED command is issued. It is suggested that you do the following:

- Notify all users before issuing the ENDTCP command so that they can end their applications.
- Issue the ENDTCP command at a time when you know no TCP/IP traffic is occurring on the system. To display the current TCP/IP traffic on the system, use option 3 on the Work with TCP/IP Status (WRKTCPSTS or NETSTAT) command.

***CNTRLD**
TCP/IP processing is ended in a controlled manner. Applications using TCP/IP are given time to complete their processing. New application processing is not allowed. After the specified period of time elapses, the processing for ENDTCP OPTION(*IMMED) is performed.

The controlled end processing does not do any of the following:

- It does not monitor to see if all TCP/IP processing has completed before the specified period of time has elapsed.
- It does not notify an application that is actively using a TCP/IP connection that TCP/IP processing will be ended.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed in which to complete a controlled end of TCP/IP processing. After this period of time all TCP/IP processing is ended immediately.

1-86400
Specify the number of seconds in which the end operation is completed.

Top

End application servers (ENDSVR)

Specifies whether or not all TCP/IP application server jobs are ended when the ENDTCP command ends TCP/IP processing.

Attention:

Before specifying *NO for this parameter, please consider the following:

- It is not possible to end all the TCP/IP processing on your system without affecting the applications which use TCP/IP.
- If TCP/IP processing is ended and no form of TCP/IP emulation (such as AnyNet) is active, then TCP/IP applications which are not restarted will not function correctly.

***YES** The ENDTCP command ends all TCP/IP application servers prior to ending TCP/IP processing.

***NO** The ENDTCP command does not end any TCP/IP application server jobs when it ends TCP/IP processing.

Note: ENDTCP ENDSVR(*NO) can be used to end TCP/IP processing without disturbing the operation of jobs using AnyNet. TCP/IP processing will be ended, however TCP/IP application servers that are using AnyNet will continue to function.

If both TCP/IP and AnyNet are inactive, use the End TCP/IP Server (ENDTCPSVR) command to end TCP/IP application server jobs.

Top

Examples

Example 1: Ending TCP/IP Immediately

```
ENDTCP   OPTION(*IMMED)
```

This command ends all TCP/IP processing on the system immediately.

Example 2: Ending TCP/IP in a Controlled Time

```
ENDTCP   OPTION(*CNTRLD)  DELAY(120)
```

This command ends all TCP/IP processing after 120 seconds have expired. During this time, new TCP/IP processing is not allowed.

Example 3: Ending TCP/IP Immediately Without Ending Application Servers

```
ENDTCP   OPTION(*IMMED)  ENDSVR(*NO)
```

This command ends all TCP/IP processing on the system immediately. However, any TCP/IP application servers (FTP, SMTP, and so on) that are active are not ended when TCP/IP processing is ended.

Top

Error messages

***ESCAPE Messages**

TCP1A13
Another job is starting or ending TCP/IP or IP over SNA.

TCP1A70
&1 not active.

TCP1A72
TCP/IP already ending with *CNTRLD option.

TCP1A73
Internal object damaged.

TCP1A74
Error occurred submitting job.

TCP1A77
&1 completed successfully; however errors occurred.

TCP9999

Internal system error in program &1.

Top

End TCP/IP Abnormal (ENDTCPABN)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End TCP/IP Abnormal (ENDTCPABN) command is used to force TCP/IP processing to terminate. It may only be used after attempting to use the End TCP/IP (ENDTCP) command with OPTION(*IMMED) specified.

The ENDTCPABN command cannot be issued until either the ENDTCP command has completed or until 10 minutes have passed following the request for TCP/IP immediate ending. This allows sufficient time for normal TCP/IP ending functions to occur.

Successful completion of ENDTCPABN processing should permit TCP/IP to be restarted without a system IPL. Issuing the ENDTCPABN command does not directly affect system termination. The next system end will **not** be marked as ABNORMAL as a result of ENDTCPABN processing.

Restrictions:

- This command is shipped with public *EXCLUDE authority. The QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles are shipped with private authorities to use this command.
- Users cannot run the ENDTCPABN command until ten minutes *after* running the ENDTCP command with OPTION(*IMMED) specified.

Top

Parameters

None

Top

Examples

```
ENDTCPABN
```

This command forces TCP/IP processing to end.

Top

Error messages

***ESCAPE Messages**

TCP1A66
ENDTCPABN not allowed at this time. Reason &1.

Top

End TCP/IP Connection (ENDTCPCNN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End TCP/IP Connection (ENDTCPCNN) command is used to end a Transmission Control Protocol/Internet Protocol (TCP/IP) connection. This command ends a connection immediately and should be used only when a normal end is not possible.

Note: The ENDTCPCNN command is usually used by specifying option 4 on the Work with TCP/IP Connection Status list of the WRKTCPSTS (NETSTAT) display. The ENDTCPCNN command is provided as a separate command to give system administrators control over this function. By limiting the authority to the ENDTCPCNN command, the system administrator limits which users can end TCP/IP connections without restricting access to the NETSTAT utility.

Top

Parameters

Keyword	Description	Choices	Notes
PROTOCOL	Protocol	*UDP, *TCP	Required, Positional 1
LCLINTNETA	Local internet address	*Character value*, *	Required, Positional 2
LCLPORT	Local port	1-65535	Required, Positional 3
RMTINTNETA	Remote internet address	*Character value*, *	Optional, Positional 4
RMTPORT	Remote port	1-65535, *	Optional, Positional 5

Top

Protocol (PROTOCOL)

Specifies the protocol used by the connection that is to be ended.

This is a required parameter.

***UDP** The connection was created for use with the User Datagram Protocol (UDP).

***TCP** The connection was created for use with the Transmission Control Protocol (TCP).

Top

Local internet address (LCLINTNETA)

Specifies the local internet address of the connection to end.

This is a required parameter.

*
The local internet address was left unspecified when this connection was opened.

Note: If '*' is specified, an attempt will be made to end both IPv4 and IPv6 connections that have an unspecified local internet address. To end only an IPv4 connection that has an unspecified local internet address, specify '0.0.0.0' (the null IPv4 address) for the LCLINTNETA value. To end only an IPv6 connection that has an unspecified local internet address, specify '::' (the null IPv6 address) for the LCLINTNETA value.

character-value
Specify the local internet address.

An IPv4 address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 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.

An IPv6 internet address is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexidecimal number ranging from 0 through X'ffff'. "::" may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The "::" may be used to compress leading, imbedded, or trailing zeros in the address. An IPv6 internet address must be a unicast address and must not contain an imbedded IPv4 address (compatibility or mapped).

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Local port (LCLPORT)

Specifies the local port number of the connection to end.

This is a required parameter.

1-65535
Specify the local port number of the connection to end.

Attention:

Ports 1 through 1024 are reserved for use by system-supplied TCP/IP applications. If the user specifies ports 1 through 1024, it can affect the operation of those applications.

Top

Remote internet address (RMTINTNETA)

Specifies the remote internet address of the connection to end. This parameter is required if PROTOCOL is *TCP.

*
The remote internet address was left unspecified when this connection was opened.

Note: If '*' is specified, an attempt will be made to end both IPv4 and IPv6 connections that have an unspecified remote internet address. To end just an IPv4 connection that has an unspecified remote internet address, specify '0.0.0.0' (the null IPv4 address) for the RMTINTNETA value. To end just an IPv6 connection that has an unspecified remote internet address, specify '::' (the null IPv6 address) for the RMTINTNETA value.

character-value
Specify the remote internet address.

An IPv4 address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 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.

An IPv6 internet address is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexidecimal number ranging from 0 through X'ffff'. ″::″ may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The ″::″ may be used to compress leading, imbedded, or trailing zeros in the address. An IPv6 internet address must be a unicast address and must not contain an imbedded IPv4 address (compatibility or mapped).

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Remote port (RMTPORT)

Specifies the remote port number of the connection to end.

This parameter is required if PROTOCOL is *TCP.

* The remote port number was left unspecified when this connection was opened.

1-65535
Specify the remote port number of the connection to end.

Top

Examples

Example 1: Ending an IPv4 TCP Connection

```
ENDTCPCNN   PROTOCOL(*TCP)  LCLINTNETA('9.5.1.109')
            LCLPORT(13054)
            RMTINTNETA('9.130.28.144')  RMTPORT(23)
```

This command ends the TCP connection between local port 13054 for local internet address 9.5.1.109 and remote port 23 for remote internet address 9.130.28.144. The TCP/IP protocol stack ends all activity on the connection and returns the resources to the free storage pools.

Example 2: Closing an IPv4 UDP Socket

```
ENDTCPCNN   PROTOCOL(*UDP)  LCLINTNETA('9.130.28.144')
            LCLPORT(596)
```

This command closes the UDP socket using local port 596 and local internet address 9.130.28.144. The TCP/IP protocol stack ends all activity on the connection and returns the resources to the free storage pools.

Example 3: Ending a LISTEN State TCP Socket

```
ENDTCPCNN   PROTOCOL(*TCP)  LCLINTNETA(*)  LCLPORT(5023)
            RMTINTNETA(*)  RMTPORT(*)
```

This command ends the TCP socket that is listening on local port 5023. The application that created this socket did not specify a local internet address. The socket is closed and the local port is made available for use by another application.

Example 4: Ending an IPv6 TCP Connection

```
ENDTCPCNN   PROTOCOL(*TCP)  LCLINTNETA('1234::5678')
            LCLPORT(13054)
            RMTINTNETA('1234::5679')  RMTPORT(23)
```

This command ends the TCP connection between local port 13054 for local internet address 1234::5678 and remote port 23 for remote internet address 1234::5679. The TCP/IP protocol stack ends all activity on the connection and returns the resources to the free storage pools.

Example 5: Closing an IPv6 UDP Socket

```
ENDTCPCNN   PROTOCOL(*UDP)  LCLINTNETA('::')
            LCLPORT(596)
```

This command closes the UDP socket using local port 596 and an unspecified IPv6 local internet address. The TCP/IP protocol stack ends all activity on the connection and returns the resources to the free storage pools.

Top

Error messages

***ESCAPE Messages**

TCP2670
Not able to complete request. TCP/IP services are not available.

TCP3B01
Not able to end TCP connection &7 &4, &8 &6.

TCP3B02
Not able to close UDP socket &7 &4.

TCP9999
Internal system error in program &1.

Top

End TCP/IP Interface (ENDTCPIFC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End TCP/IP Interface (ENDTCPIFC) command ends a Transmission Control Protocol/Internet Protocol (TCP/IP) interface. When an interface is ended with this command, datagrams addressed to the IP addresses associated with this interface will no longer be accepted. However, the operation of any other TCP/IP or IP over SNA interface that is using the same line description as the the interface being ended is not affected.

This command can be used to end an interface that was previously started by the Start TCP/IP Interface (STRTCPIFC) or Start TCP/IP (STRTCP) command.

Warning: Temporary Level 2 Header

Warning: Temporary Level 3 Header

Notes on Route to Interface Binding

Interfaces define direct routes to networks or subnetworks that this system is directly attached to. Routes define the indirect routes. An indirect route defines the next hop on the path to a network or subnetwork that this system is not directly attached to.

Indirect routes are bound to interfaces using a best match first algorithm. This algorithm is based on the state of the interface and on the type of service (TOS) specified for the route and interface. When ending an interface, the routes associated with the interface can move to another existing active interface. This provides the widest available level of connectivity.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*, *IP6SAC	Optional, Positional 1
ALIASNAME	Alias name	*Simple name*	Optional
LIND	Line description	*Name*, ***ADRALIAS**	Optional

Top

Internet address (INTNETADR)

Specifies the internet address of an interface that had previously been added to the TCP/IP configuration with the Add TCP/IP Interface (ADDTCPIFC) command and which had been previously started by the STRTCPIFC or STRTCP command.

Note: Either the INTNETADR or the ALIASNAME parameter must be specified for the command, but not both of them.

***IP4DHCP**

Specify that Dynamic Host Configuration Protocol should end for the associated line description. A line description name must be specified for the LIND parameter.

***IP6SAC**

Specify that IPv6 stateless address auto-configuration should end for the associated line description. All auto-configured IPv6 interfaces will end. A line description name must be specified for the LIND parameter.

character-value

Specify the internet address associated with the interface to be ended.

An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

An IPv6 internet address is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexadecimal number ranging from 0 through X′FFFF′. ″::″ may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The ″::″ may be used to compress leading, imbedded, or trailing zeros in the address.

An IPv6 internet address must be a unicast address and must not contain an imbedded IPv4 address (compatibility or mapped). If the IPv6 address is a link-local unicast address, a line description name must be specified for the LIND parameter.

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Alias name (ALIASNAME)

Specifies the name of the interface to be ended. The alias name specified may be for an IPv4 or IPv6 interface.

Note: Either the INTNETADR or the ALIASNAME parameter must be specified for the command, but not both of them.

simple-name

Specify the name of the interface to be ended.

Top

Line description (LIND)

Specifies the name of the line description associated with the interface being ended.

***ADRALIAS**

The line description is determined from either the INTNETADR value or the ALIASNAME value.

name Specify the name of the line description associated with the interface being ended. The line description must be specified if the INTNETADR value is *IP4DHCP, *IP6SAC, or an IPv6 link-local unicast address.

Top

Examples

Example 1: Ending an X.25 Interface

```
ENDTCPIFC   INTNETADR('9.5.11.125')
```

This command causes the TCP/IP protocol stack to deactivate (end) the interface associated with the internet address 9.5.11.125.

Example 2: Ending a Token-Ring Interface

```
ENDTCPIFC   INTNETADR('156.93.81.7')
```

This command causes the TCP/IP protocol stack to deactivate (end) the interface associated with the internet address 156.93.81.7.

Example 3: Ending a Token-Ring Interface using an Alias Name

```
ENDTCPIFC   ALIASNAME(TEST_NETWORK)
```

This command causes the TCP/IP protocol stack to deactivate (end) the interface associated with the alias name TEST_NETWORK.

Example 4: Ending an Ethernet IPv6 Interface

```
ENDTCPIFC   INTNETADR('1234:5678:9ABC:DEF0:1111:2222:3333:4444')
```

This command causes the TCP/IP protocol stack to end the IPv6 interface associated with address 1234:5678:9ABC:DEF0:1111:2222:3333:4444.

Example 5: Ending an Ethernet IPv6 Link-Local Interface

```
ENDTCPIFC   INTNETADR('FE80::1234') LIND(ETHLINE)
```

This command causes the TCP/IP protocol stack to end the IPv6 link-local interface associated with address FE80::1234 and line description ETHLINE.

Example 6: Ending IPv6 Stateless Address Auto-Configuration On an Ethernet Line

```
ENDTCPIFC   INTNETADR(*IP6SAC) LIND(ETHLINE2)
```

This command causes the TCP/IP protocol stack to end IPv6 stateless address auto-configuration for line ETHLINE2.

Top

Error messages

***ESCAPE Messages**

TCP1B15
Line description &2 unusable. Internal errors encountered.

TCP1B61
Unable to determine if &1 interface ended.

TCP1B62
Cannot determine if &1 interface ended.

TCP1B65
&2 interface not ended. Reason &1.

TCP1B71
&1 interface not ended.

TCP1B72
&1 interface not ended. &1 interface is not active.

TCP1B73
&1 interface not ended. &1 interface not defined in the TCP/IP configuration.

TCP1B74
&1 interface not ended. Line description &2 not found.

TCP1B85
Unable to submit request to end interface &1.

TCP265F
INTNETADR parameter value &2 not valid.

TCP9999
Internal system error in program &1.

Top

End Point-to-Point TCP/IP (ENDTCPPTP)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End Point-to-Point TCP/IP (ENDTCPPTP) command is used to end a point-to-point TCP/IP session job. A session job operates in one of two possible modes:

1. Answer mode sessions (*ANS) allow a remote system to contact the local system and establish a point-to-point TCP/IP session.
2. Dial mode sessions (*DIAL) allow the local system to contact a remote system that supports point-to-point TCP/IP.

Note: Profiles of linetype *PPP can ended with this command but any configuration of *PPP profiles must be done using the System i Navigator graphical user interface.

The TCP/IP point-to-point session jobs run in the QSYSWRK subsystem.

Top

Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	*Character value*, *ALL	Required, Positional 1
OPRMODE	Operating mode	***ANY***, *ANS, *DIAL	Optional, Positional 2

Top

Configuration profile (CFGPRF)

Specifies which point-to-point TCP/IP sessions job or jobs should be ended.

This is a required parameter.

***ALL** All currently active point-to-point TCP/IP sessions jobs operating in the mode specified by the OPRMODE parameter are ended.

generic-name
Specify the generic name of the point-to-point TCP/IP configuration profile to be ended. 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 profiles with names that begin with the generic name are ended. If an asterisk is not included, the name is assumed to be a complete point-to-point TCP/IP configuration profile name. All currently active point-to-point TCP/IP session jobs using the profiles indicated and operating in the mode specified by the OPRMODE parameter are ended.

name Specify the name of a TCP/IP point-to-point configuration profile. The active point-to-point session job using this profile is ended.

Top

Operating mode (OPRMODE)

Specifies the operating mode of the TCP/IP point-to-point session job to be ended.

***ANY** Any point-to-point TCP/IP session job that matches the configuration profile name specified on the CFGPRF parameter is ended, regardless of operating mode.

***ANS** The operating mode of the session to be ended is *ANS. All *ANS point-to-point TCP/IP session jobs that are currently active that match the specified CFGPRF parameter will be ended.

***DIAL**
The operating mode of the session to be ended is *DIAL. All *DIAL point-to-point TCP/IP session jobs that are currently active that match the specified CFGPRF parameter will be ended.

Top

Examples

Example 1: End a TCP/IP Point-To-Point Session Job

```
ENDTCPPTP   CFGPRF(DIALPRF)
```

This command ends the point-to-point TCP/IP session job that is using configuration profile DIALPRF. The operating mode (OPRMODE) value will default to *ANY so the operating mode is not used in deciding whether to end the session job.

Example 2: End All Answer (*ANS) Mode TCP/IP Point-To-Point Session Jobs

```
ENDTCPPTP   CFGPRF(*ALL)  OPRMODE(*ANS)
```

This command ends all active or activating point-to-point answer mode (*ANS) TCP/IP session jobs.

Example 3: End All TCP/IP Point-To-Point Session Jobs

```
ENDTCPPTP   CFGPRF(*ALL)
```

This command ends all active or activating point-to-point TCP/IP session jobs.

Example 4: End All TCP/IP Point-To-Point Session Jobs Starting with XYZ.

```
ENDTCPPTP   CFGPRF(XYZ*)
```

This command ends all active or activating point-to-point TCP/IP session jobs that have profiles that begin with XYZ.

Example 5: End an Answer Mode TCP/IP Point-To-Point Session Job using a Specific Profile Name

```
ENDTCPPTP   CFGPRF(DIALPRF)  OPRMODE(*ANS)
```

This command will end the point-to-point TCP/IP session job using profile DIALPRF if this profile is defined to run in answer mode. If the profile is defined to run in dial mode then no action will be taken.

Top

Error messages

***ESCAPE Messages**

TCP1A1F
Cannot process request while &3/&2/&1 using &6.

TCP8205
Required object &2/&1 type *&3 not found.

TCP8209
ENDTCPPTP &1 &3 for job &6/&5/&4 completed. &10 of &11 sessions ended.

Top

End TCP/IP Server (ENDTCPSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The End TCP/IP Server (ENDTCPSVR) command is used to end the TCP/IP application server jobs that are specified in the SERVER parameter. If the jobs have any current active connections, these connections are ended immediately. If the ENDTCPSVR command is used to end a server that is not active, a diagnostic message may be returned.

The End TCP/IP Server command can only be used when TCP/IP is fully operational. When the system is in restricted state, this command is not allowed.

Additional servers can automatically be added to the list of servers that ENDTCPSVR will support by using the Add TCP/IP Server (ADDTCPSVR) command.

Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server application	Single values: ***ALL** Other values (up to 300 repetitions): *Character value*	Optional, Positional 1
HTTPSVR	HTTP server	Single values: ***ALL** Other values: *Element list*	Optional
	Element 1: Server instance	*Name*, *ADMIN	
DNSSVR	DNS server	Single values: ***ALL** Other values: *Element list*	Optional
	Element 1: Server instance	*Name*	
TCMSVR	TCM server	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Instance name	*Character value*, *ALL	
TOMCATSVR	ASFTOMCAT server	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Server instance name	*Character value*, *ALL	
INSTANCE	Instance	*Character value*, ***DFT**, *ALL	Optional

Top

Server application (SERVER)

Specifies which of the TCP/IP application server jobs is to be ended by this command.

Additional TCP/IP servers could also be available if they were added by running the Add TCP/IP Server (ADDTCPSVR) command.

For a list of all supported values for this parameter, you can press F4 (Prompt) for this parameter when prompting this command.

Single values

***ALL** All of the TCP/IP server jobs are ended.

Other values (up to 300 repetitions)

***BOOTP**
The bootstrap protocol (BOOTP) server is ended.

***DBG** The Debug server is ended.

***DDM**
The Distributed Data Management (DDM) server job is ended.

***DHCP**
The Dynamic Host Configuration Protocol (DHCP) server job is ended.

***DIRSRV**
The LDAP directory services (DIRSVR) server job is ended.

***DLFM**
The DataLink File Manager (DLFM) server job is ended.

***DNS** The Domain Name System (DNS) server job is ended unless:

- A specific server instance name is specified for the DNSSVR parameter. Then only the specified instance is ended.

To end all instances of the DNS server, specify one of the following:

```
ENDTCPSVR   SERVER(*DNS)
ENDTCPSVR   SERVER(*DNS)  DNSSVR(*ALL)
```

***DOMINO**
The Lotus Domino (DOMINO) server is ended.

***EDRSQL**
The Extended Dynamic Remote SQL (EDRSQL) server is ended.

***FTP** All File Transfer Protocol (FTP) server jobs are ended.

***HOD** The Host On Demand (HOD) server is ended.

***HTTP**
All instances of the HyperText Transfer Protocol (HTTP) server are ended unless:

- A specific server instance name is specified for the HTTPSVR parameter. Then only the specified instance is ended.
- The *ADMIN special value is specified for the HTTPSVR parameter. Then only the administration server is ended.

To end all instances of the HTTP server, specify one of the following:

```
ENDTCPSVR   SERVER(*HTTP)
ENDTCPSVR   SERVER(*HTTP)  HTTPSVR(*ALL)
```

The HTTP server is also known as the IBM HTTP Server.

***IBMHELP**
The IBM Online Help and Eclipse Information Center (IBMHELP) server is ended.

***INETD**
The Internet Daemon (INETD) server is ended.

***LPD** All line printer daemon (LPD) servers are ended.

***MGTC**
The Management Central (MGTC) server is ended.

***NETSVR**
The NetServer (NETSVR) server is ended.

***NSLD**
The Network Station Login Daemon (NSLD) server is ended.

***NTP** All Simple Network Time Protocol (SNTP) services servers are ended.

Note: If both client and server SNTP services have been started, running the ENDTCPSVR command specifying SERVER(*NTP) will end both client and server services. If you wanted to end just the client or just the server SNTP service, you will need to run the Start TCP/IP Server (STRTCPSVR) command again, specifying SERVER(*NTP) and NTPSRV(*CLIENT) or NTPSRV(SERVER).

***ODPA**
The On-Demand Platform Authentication (ODPA) server is ended.

***OMPROUTED**
The OMPROUTER Daemon (OMPROUTED) server is ended. The Open Shortest Path First (OSPF) and Routing Information Protocol (RIP) server jobs are ended. Specifying SERVER(*OMPROUTED) and INSTANCE(*OSPF) will end only the OSPF server job. Specifying SERVER(*OMPROUTED) and INSTANCE(*RIP) will end only the RIP server job.

***ONDMD**
The OnDemand (ONDMD) server job is ended.

***POP** All Post Office Protocol (POP3) mail server jobs are ended.

***QOS** The Quality of Service (QOS) server is ended.

***REXEC**
All Remote Execution (REXEC) servers are ended.

***ROUTED**
The Router Daemon (ROUTED) server is ended.

***SMTP**
All jobs associated with Simple Mail Transfer Protocol (SMTP) in the QSYSWRK subsystem are ended. The bridge job in the QSNADS subsystem is not ended.

***SNMP**
All jobs associated with the Simple Network Management Protocol (SNMP) agent in the QSYSWRK subsystem are ended.

***SRVSPTPRX**
The Service and Support Proxy (SRVSPTPRX) server is ended.

***SSHD**
The Secure Shell (SSH) Daemon server is ended.

***TCM** The Triggered Cache Manager (TCM) server is ended unless:

- A specific server instance name is specified for the TCMSVR parameter. Then only the specified instance is ended.

To end all instances of the TCM server, specify:

```
ENDTCPSVR   SERVER(*TCM)  TCMSVR(*ALL)
```

***TELNET**
All TELNET server jobs are ended.

***TFTP** All Trivial File Transfer Protocol (TFTP) server jobs are ended.

***VPN** The Virtual Private Network (VPN) server is ended.

***WEBFACING**
The WebFacing server is ended.

Top

HTTP server (HTTPSVR)

Specifies the name of the HTTP server instance to end. The SERVER parameter specified must be *HTTP or this parameter is ignored.

If multiple HTTP server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***ALL** All instances of the HTTP server that are currently running are ended.

***ADMIN**
The Administration Server is ended. The Administration Server is an instance of the HTTP server that allows administration of certain system functions using a Web browser.

name Specify the name of the HTTP server instance to be ended.

Top

DNS server (DNSSVR)

Specifies the name of the DNS server instance to end. The SERVER parameter specified must be *DNS or this parameter is ignored.

If multiple DNS server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***ALL** All instances of the DNS server that are currently running are ended.

name Specify the name of the DNS server instance to be ended.

Top

TCM server (TCMSVR)

Specifies the name of the TCM server instance to end. The SERVER parameter specified must be *TCM or this parameter is ignored.

If multiple TCM server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***NONE**
No instances of the TCM server that are currently running are ended.

***ALL** All instances of the TCM server that are currently running are ended.

character-value
Specify the name of the TCM server instance to be ended.

Top

ASFTOMCAT server (TOMCATSVR)

Specifies the name of the Tomcat server instance to end. The SERVER parameter specified must be *ASFTOMCAT or this parameter is ignored.

If multiple Tomcat server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***NONE**
No instances of the Tomcat server that are currently running are ended.

***ALL** All instances of the Tomcat server that are currently running are ended.

character-value
Specify the name of the Tomcat server instance to be ended.

Top

Instance (INSTANCE)

Specifies the name of the server instance to end. This parameter may be specified if the SERVER parameter is *DIRSRV, *DNS, *HTTP, *OMPROUTED, or *TCM. It may also be specified for user-defined servers which support multiple server instances.

***DFT** The default server or servers are ended. If the SERVER parameter is *DIRSRV, the QUSRDIR server instance will be ended. If the SERVER parameter is *HTTP or *DNS, all defined server instances will be ended. If the SERVER parameter is *OMPROUTED, both the *OSPF and *RIP server jobs are ended. If the SERVER parameter is *TCM, no defined server instances will be ended

***ALL** All instances of the server that are currently running are ended.

character-value
Specify the name of the server instance to be ended. Up to 32 characters may be specified.

Top

Examples

Example 1: Ending All TCP/IP Servers

```
ENDTCPSVR   SERVER(*ALL)
```

This command ends all active TCP/IP application server jobs.

Example 2: Ending the LPD Servers

```
ENDTCPSVR   SERVER(*LPD)
```

This command ends the TCP/IP LPD application server jobs.

Example 3: Ending a Specific HTTP Server Instance

```
ENDTCPSVR   SERVER(*HTTP)  HTTPSVR(HTTP1)
ENDTCPSVR   SERVER(*HTTP)  INSTANCE(HTTP1)
```

These commands end the TCP/IP HTTP application server instance named HTTP1.

Example 4: Ending a Specific DNS Server Instance

```
ENDTCPSVR   SERVER(*DNS)  DNSSVR(DNS1)
ENDTCPSVR   SERVER(*DNS)  INSTANCE(DNS1)
```

These commands end the TCP/IP DNS application server instance named DNS1.

Example 5: All Directory Services Server Instances

```
ENDTCPSVR   SERVER(*DIRSRV) INSTANCE(*ALL)
```

This command ends all instances of the Directory Services server.

Top

Error messages

***ESCAPE Messages**

CPF3894
Cancel reply received for message &1.

TCP1A0A
&1 ended abnormally. Reason code is &2.

TCP1A11
&1 failed.

TCP1A77
&1 completed successfully; however errors occurred.

Top

End TIE Session (ENDTIESSN)

Where allowed to run:

- Batch job (*BATCH)
- Batch program (*BPGM)
- Batch REXX procedure (*BREXX)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The End Technical Information Exchange Session (ENDTIESSN) command allows you to disconnect the communications line used for TIE batch commands. This command must follow other TIE batch commands.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
ENDTIESSN
```

This command ends the TIE function by disconnecting the communications line used for TIE batch commands.

Top

Error messages

None

Top

End Trace (ENDTRC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Trace (ENDTRC) command ends a trace session that was started by a STRTRC (Start Trace) command.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service trace function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, 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.
- If DTAOPT(*LIB) is specified, you must have authority to the library and the database files within that library where the trace data is stored.
- If PRTTRC(*YES) is specified, you must have authority to the PRTTRC (Print Trace) command.

Top

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	*Name*, *PRV	Required, Positional 1
OPTION	Option	***END**, *EXTRACT	Optional
DTAOPT	Data option	***LIB**, *DLT	Optional
DTALIB	Data library	*Name*, ***CURLIB**	Optional
RPLDTA	Replace data	***YES**, *NO	Optional
SLTJOB	Select jobs	Single values: ***ALL** Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Job name	*Qualified job name*	
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000001-999999	
	Element 2: Thread ID to include	Single values: ***ALL** Other values (up to 20 repetitions): *Character value*, *INITIAL	
PRTTRC	Print trace data	***NO**, *YES	Optional

Top

Session ID (SSNID)

Specifies a session identifier for the trace to be ended. This name must match the session identifier of a trace that had been previously started and is still active.

This is a required parameter.

***PRV** The trace session most recently started by the same user who is running this ENDTRC command will be ended. For example, if the job running the ENDTRC command is running under user profile BOB, the last trace session started under user profile BOB is ended.

name Specify the session identifier of the trace to be ended.

Top

Option (OPTION)

Specifies if trace data will be preserved in the internal buffers where it was collected after the session is ended.

***END** The trace data will not be preserved in the internal buffers where it was collected. After session is ended, the session can be:

- Printed with PRTTRC command if DTAOPT(*LIB) is specified.

***EXTRACT**

The trace data will be preserved in the internal buffers where it was collected. After data is extracted, the session can be:

- Printed with PRTTRC command.
- Extracted again with ENDTRC OPTION(*EXTRACT) command.
- Ended with ENDTRC OPTION(*END) command.

Top

Data option (DTAOPT)

Specifies if the trace data that has been collected is stored into database files.

***LIB** The trace data will be copied into database files. The PRTTRC parameter on this command or the Print Trace (PRTTRC) command can be used to format and print the data.

***DLT** The trace data will not be copied into database files and the trace data will be deleted from the internal buffers where it was collected.

Note: This option is only valid if *END was specified for the **Option (OPTION)** parameter.

Top

Data library (DTALIB)

Specifies the name of the library in which the trace data will be stored. A set of database files will be created in this library to contain the trace data. The files will be created if they do not already exist.

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

***CURLIB**

The trace data is stored in files in the current library for the job. If no library is specified as the current library for the job, QGPL is used.

name Specify the name of the library to contain the trace data files. The library must already exist when the ENDTRC command is run.

Top

Replace data (RPLDTA)

Specifies whether trace data that was collected by a previous trace session with the same session identifier is replaced with new trace data. This is determined by checking if the set of database files where the trace data is to be stored already have file members with the same name as the specified trace session identifier (SSNID parameter).

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

***YES** If trace data already exists with the specified session identifier, the old trace data is lost and replaced by the new trace data.

***NO** If trace data already exists for the specified session, an error message is sent to the user.

Top

Select jobs (SLTJOB)

Specifies which jobs to include in the data collection. This allows the user to reduce the size of the trace data by selecting only a subset of the jobs that were part of the collection. Up to 8 qualified job names can be specified.

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

Single values

***ALL** All jobs that were part of the data collection are included.

Other values (up to 8 repetitions)

Element 1: Job name

Qualifier 1: Job name

name Specify the name of the job to be included in the data collection.

Qualifier 2: User

name Specify the name of the user of the job to be included in the data collection.

Qualifier 3: Number

000001-999999
Specify the number of the job to be included in the data collection.

Element 2: Thread ID to include

Single values

***ALL** All threads of the specified job are included.

Other values (up to 20 repetitions)

***INITIAL**
Only the initial thread of the specified job is included.

thread-identifier

Specify the thread identifier of the job to be included. This is the thread ID as shown by the Work with Job (WRKJOB) command.

Top

Print trace data (PRTTRC)

Specifies whether trace data is formatted and printed after it is stored in the trace database files.

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

<u>*NO</u> The PRTTRC (Print Trace) command is not run as part of this command.

***YES** The PRTTRC (Print Trace) command is run after the trace data has been stored in the database files.

Top

Examples

Example 1: End Most Recently Started Trace

```
ENDTRC   SSNID(*PRV)
```

This command ends the trace session started most recently by the same user who is running the ENDTRC command. The trace data will be stored in a set of files in the current library of the job, or QGPL if there is no current library for the job.

Example 2: End a Trace and Delete Trace Data

```
ENDTRC   SSNID(DCG1)
         OPTION(*END)
         DTAOPT(*DLT)
```

This command ends the trace session DCG1 and deletes the trace data.

Example 3: End a Trace Session Extracting a Specific Job to be Collected

```
ENDTRC   SSNID(MYTRACE)
         OPTION(*EXTRACT)
         SLTJOB((123456/MYUSER/MYJOBNAME))
```

This command ends the trace session MYTRACE; only the 123456/MYUSER/MYJOBNAME job is collected. The session might be extracted again later.

Top

Error messages

<u>*ESCAPE Messages</u>

CPF39CA

Trace session ID &1 not found.

CPF39CB

Trace session ID &1, in library &2, data exists. Specify RPLDTA(*YES).

CPF98A2

Not authorized to &1 command or API.

CPF39D3

Unable to start/end the trace.

Top

End Trap Manager (ENDTRPMGR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

Use the End Trap Manager (ENDTRPMGR) command to end the i5/OS Simple Network Management Protocol (SNMP) trap manager job.

Top

Parameters

None

Top

Examples

```
ENDTRPMGR
```

This command ends the i5/OS SNMP Manager Framework trap manager job.

Top

Error messages

***ESCAPE Messages**

CPFA805
Trap manager job not active or being ended.

Top

End Watch (ENDWCH)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Watch (ENDWCH) command ends a watch session that was started by a STRWCH (Start Watch) command or by Start Watch (QSCSWCH) API. Watch sessions started by trace commands (STRTRC, TRCINT, TRCCNN, STRCMNTRC, TRCTCPAPP) will be ended but the associated trace will remain active.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service watch function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_WATCH, can also be used to change the list of users that are allowed to start and end watch operations.
- If ending a watch session that is watching for a message within a job log, the issuer of the command must be running under a user profile which is the same as the job user identity of the job being watched, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. Job control (*JOBCTL) special authority is also required when ending a session where jobs with a generic user name are being watched.
- If ending a watch session that was started specifying *ALL for the watch job name, or a generic user name, you must have *ALLOBJ special authority, or be authorized to the Watch any job function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_WATCH_ANY_JOB, can also be used to change the list of users that are allowed to start and end watch operations.

Top

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	*Name*, *PRV	Required, Positional 1

Top

Session ID (SSNID)

Specifies a session identifier for the watch to be ended. This name must match the session identifier of a watch session that had been previously started and is still active.

This is a required parameter.

***PRV** The watch session most recently started by the same user who is running this ENDWCH command will be ended. For example, if the job running the ENDWCH command is running under user profile BOB, the last watch session started under user profile BOB is ended.

name Specify the session identifier of the watch to be ended.

Top

Examples

Example 1: End Most Recently Started Watch

```
ENDWCH   SSNID(*PRV)
```

This command ends the watch session started most recently by the same user who is running the ENDWCH command.

Example 2: End a Specific Watch Session

```
ENDWCH   SSNID(MYSESSION)
```

This command ends the watch session MYSESSION.

Top

Error messages

***ESCAPE Messages**

CPF39E1
Watch session &1 not found.

CPF39E2
There is no active watch session for current user profile.

CPF39E6
The user does not have the required authority.

CPF39E8
Not enough authority to watch operations.

CPF39E9
*JOBCTL special authority required.

Top

End Writer (ENDWTR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The End Writer (ENDWTR) command ends the specified spooling writer and makes its associated output device available to the system. The writer can be ended immediately or in a controlled manner. If ended immediately, the writer stops writing the file and the file is made available again on the output queue. If ended in a controlled manner, the writer finishes writing the current file (or a copy of a file), or it finishes printing a page of the file, before it is ended.

Top

Parameters

Keyword	Description	Choices	Notes
WTR	Writer	*Name*, *SYSVAL, *ALL	Required, Positional 1
OPTION	When to end writer	***CNTRLD**, *IMMED, *PAGEEND	Optional, Positional 2

Top

Writer (WTR)

Specifies the spooling writer being stopped. The writer's output device will then be available to the system.

This is a required parameter.

***ALL** Specifies that all writers that are started are to stop.

***SYSVAL**
Specifies that the writer started to the system default printer is to stop.

name Specify the name of the writer to end.

Top

When to end writer (OPTION)

Specifies when the writer should stop processing.

***CNTRLD**
The spooling writer stops processing in a controlled manner. Output stops at the end of the spooled file (or copy of a file) currently being written to an output device.

***IMMED**
The writer stops processing immediately. The spooled file that is currently printing remains on the output queue.

***PAGEEND**
The writer is stopped after processing of the current buffer. This value is valid only if the spooling writer is a printer writer.

Top

Examples

```
ENDWTR   WTR(PRINTER)
```

This command stops the writer named PRINTER at the end of the spooled file whose output is being printed, and then releases the device to the system.

Top

Error messages

***ESCAPE Messages**

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1340
Job control function not performed.

CPF1352
Function not done. &3/&2/&1 in transition condition.

CPF1842
Cannot access system value &1.

CPF3313
Writer &1 not active nor on job queue.

CPF3330
Necessary resource not available.

CPF3331
Not authorized to control writer &3/&2/&1.

CPF3339
Previous end request to writer &3/&2/&1 pending.

CPF3438
*PAGEEND not valid for writer &3/&2/&1.

Top

Remove Link (ERASE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Remove link (ERASE) command removes the link to the specified object. If this is the only hard link to the object, the object is removed when no longer in use. The object can be removed even if a symbolic link to it exists. The symbolic link remains until it is removed.

This command is an alias for the Remove link (RMVLNK) command and can also be issued using the following alternative command names:

- DEL
- RMVLNK

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

- In the ″root″ (/), QOpenSys, and user-defined file systems, the user must have write and execute (*WX) authority to the directory containing the object. If a hard link is to be unlinked, the user must also have object existence (*OBJEXIST) authority to the object.
- In the QDLS file system, the user must have all (*ALL) authority to the object and execute (*X) authority to the parent directory.
- The user must have *X authority to each directory in the path.
- See the System i Security Reference, SC41-5302 book for the authority requirements for other file systems.
- A user cannot unlink an object within a ″root″ (/), QOpenSys, or user-defined file system directory that has the ″restricted rename and unlink″ attribute set on (this attribute is equivalent to the S_ISVTX mode bit) unless one or more of the following are true:
  - The user is the owner of the object.
  - The user is the owner of the directory.
  - The user has all object (*ALLOBJ) special authority.
- A directory cannot be unlinked.
- The link to a file cannot be removed if the file is a DataLink column in an SQL table and where a row in that SQL table references this file.
- The restrictions listed above are for the i5/OS objects of the types *DDIR, *DSTMF, *SOCKET, *STMF, and *SYMLNK.

QSYS.LIB and independent ASP QSYS.LIB File System Differences

- If this command is to be used to remove links for an object that is in these file systems, additional restrictions may apply. To identify these restrictions, see the delete command for the object to be removed. In general, the name of this command is formed using the i5/OS object type value, from the character * is removed, and add the verb DLT to the beginning. For example, to delete an alert table, which has the object type value of *ALRTBL, see the Delete Alert Table (DLTALRTBL) command for any additional restrictions.

  However, there are exceptions to this rule. For example, to delete a compiler unit, which has the object type value of *MODULE, see the Delete Module (DLTMOD) command for any additional restrictions.

For a description of the object types, see the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

- In these file systems, libraries and database files cannot be deleted using the Remove Link (RMVLNK or alias DEL or ERASE) command. However, these objects can be deleted using the Remove Directory (RMVDIR or alias RMDIR or RD) command.
- The following object types cannot be deleted using another command: *EXITRG, *IGCSRT, *JOBSCD, *PRDAVL, *QRYDFN, *RCT.

QDLS File System Differences

- If this command is to be used to remove links for an object that is in this file system, additional restrictions may apply. To identify these restrictions, see the description of the Delete Document Library Object (DLTDLO) command.

Top

Parameters

Keyword	Description	Choices	Notes
OBJLNK	Object link	*Path name*	Required, Positional 1

Top

Object link (OBJLNK)

Specifies the path name of the object to unlink. Multiple links can be removed with a name pattern.

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

For more information on specifying path names, refer to ″Object naming rules″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Top

Examples

The alternative command name for ERASE is RMVLNK. The following examples use the alternative command name, but ERASE can be replaced directly for RMVLNK in all of them.

Example 1: Removing an Object Link

```
RMVLNK   OBJLNK('PAY')
```

This command removes a link named PAY.

Top

Error messages

***ESCAPE Messages**

CPFA085
Home directory not found for user &1.

CPFA093
Name matching pattern not found.

CPFA09C
Not authorized to object. Object is &1.

CPFA0A1
An input or output error occurred.

CPFA0A7
Path name too long.

CPFA0A9
Object not found. Object is &1.

CPFA0AB
Operation failed for object. Object is &1.

CPFA0B1
Requested operation not allowed. Access problem.

CPFA0B2
No objects satisfy request.

CPFA0BD
&1 links removed. &2 links failed.

Top

Export a Program Symbol (EXPORT)

Parameters
Examples
Error messages

The Export a Program Symbol (EXPORT) binder definition statement defines an export in a service program export block.

Top

Parameters

Keyword	Description	Choices	Notes
SYMBOL	Exported symbol name	*Character value*	Required, Positional 1

Top

Exported symbol name (SYMBOL)

Specifies the symbol to be exported. The symbol can be enclosed in apostrophes, quotation marks, or expressed without the delimiting marks.

This is a required parameter.

character-value
Specify the name of the program external variable or procedure to be exported.

Top

Examples

```
EXPORT  SYMBOL('ExtVar2')
```

This binder definition statement defines **ExtVar2** as an exported symbol in a service program export block.

Top

Error messages

None

Top

Change NFS Export (EXPORTFS)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Change Network File System Export (EXPORTFS) command adds directory names to (exports) or removes directory names from (unexports) the list of directory trees that are currently exported to Network File System (NFS) clients. The flags in the OPTIONS list indicate what actions the EXPORTFS command should perform.

A list of directories and options for exporting the directory and its contents is stored in the */etc/exports* file. The EXPORTFS command allows the user to export all of the directory trees specified in the */etc/exports* file using the -A flag, or to export a single directory tree by specifying the directory name. When the directory tree to be exported exists in the */etc/exports* file, the user can export it with the options specified there, or one can use the -I flag to override the options, specifying the new options on the EXPORTFS command.

Ths user can also export a directory tree not previously defined in the */etc/exports* file by providing the options for it on the EXPORTFS command. The user can unexport directory trees by using the -U flag on the EXPORTFS command.

The forced unexport will be available only if the user specifies the -U flag and wants to release all NFS version 4 locks and state held by the NFS server on the exported file system. This option should only be used if the state cannot be removed through normal methods.

The user can also add, change, or remove export entries in the */etc/exports* file by using the -F flag.

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

- CHGNFSEXP

For more information about Network File System commands, see the i5/OS Network File System Support book, SC41-5714.

Restrictions:

- The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
- The user must have execute (*X) authority to each directory in the path name prefixes.
- When the -F flag is specified and the */etc/exports* file does not exist, the user must have write and execute (*WX) authority to the */etc* directory.
- When the -F flag is specified and the */etc/exports* file does exist, the user must have read and write (*RW) authority to the */etc/exports* file and *X authority to the */etc* directory.
- Mixed CCSID encoding schemes are not supported. Specified CCSIDs must be single-byte character set (SBCS) or pure double-byte character set (DBCS).

Top

Parameters

Keyword	Description	Choices	Notes
OPTIONS	NFS export options	*Character value*, ***DFT**	Optional, Positional 1
DIR	Directory	*Path name*	Optional, Positional 2
HOSTOPT	Host options	Single values: ***DFT** Other values (up to 10 repetitions): *Element list*	Optional
	Element 1: Host name	*Character value*	
	Element 2: Data file CCSID	0-65535, ***BINARY**, *ASCII, *JOBCCSID	
	Element 3: Path name CCSID	0-65535, ***ASCII**, *JOBCCSID	
	Element 4: Force synchronous write	***SYNC**, *ASYNC	

Top

NFS export options (OPTIONS)

The export options list contains some flags followed optionally by a list containing a character string of characteristics for the directory tree to be exported.

Each flag consists of a minus ″-″ followed by a character. The flags are separated by spaces. Only certain combinations of flags are allowed. If an invalid combination is detected, an error is returned.

Note: A value (other than *NONE) must be specified for either the OPTIONS or **Directory (DIR)** parameter. Both OPTIONS and DIR can be specified so long as ′-A′ is not part of the options list specified for the OPTIONS parameter.

***DFT**
The default value for the options string is:

```
'-A'
```

options-flags

-A
This is the ″all″ flag and it indicates that all entries in the */etc/exports* file are to be processed. The following flag combinations have special significance:

-A and not -U
This will export every entry in the */etc/exports* file (making them available to NFS clients).

-A and -U
This will unexport every entry that is currently exported (making them unavailable to NFS clients). This makes no reference to the contents of the */etc/exports* file.

-A and the DIR parameter
This combination is not allowed.

-A and (-I or -F or -O)
These combinations are not allowed.

-I
This is the ″ignore″ flag and it indicates, for the directory tree specified in the DIR parameter, how the export characteristics are determined. The following flag combinations have special significance:

-I and -O
The export characteristics specified on the -O flag are used, and the definitions listed in the */etc/exports*, if they exist, are ignored.

not -I and not -O
Either the export characteristics listed for the entry in the */etc/exports* file are used, or, if there are no options in that file, the default options are assumed. See the -O flag description for the list of default options.

-I and (-A or -U)
These combinations are not allowed.

-U This is the ″unexport″ flag and it indicates that the specified directory tree entered in the DIR parameter is to be unexported (made unavailable to NFS clients). The following flag combinations have special significance:

FORCE
Specifies a forced unexport of the entry referenced in the DIR parameter; this option will only be available if the -U flag has been specified. The FORCE option will release all NFS version 4 locks and state for the exported path. This option should only be used when the locks and state for the exported path cannot be released though normal methods.

-U and -A
This will unexport every entry that is currently exported (making them unavailable to NFS clients). This makes no reference to the contents of the */etc/exports* file.

-U and -F
The entry referenced in the DIR parameter is removed from the */etc/exports* file (if it exists) in addition to being unexported (making it unavailable to NFS clients).

-U and (-I or -O)
These combinations are not allowed.

-F This is the ″file″ flag and it requires the DIR parameter. The following flag combinations have special significance:

-F and -U
The entry referenced in the DIR parameter is removed from the */etc/exports* file (if it exists) in addition to being unexported (making it unavailable to NFS clients).

-F and not -U and not -O
The specified directory tree entered in the DIR parameter is to be exported (made available to NFS clients). In addition, an entry for this directory tree entered in the DIR parameter will be added to the */etc/exports* file. If the entry already exists in the file, it will be replaced with this new export entry. If the file does not exist, it will be created and the export entry will be added to it. Note that the ″ignore″ flag -I is implied when the ″file″ flag -F is specified without the ″unexport″ flag -U. Since the ″options″ flag -O is not specified, the default options are assumed. See the -O flag description for the list of default options.

-F and not -U and -O
The specified directory tree entered in the DIR parameter is to be exported (made available to NFS clients). In addition, an entry for this directory tree entered in the DIR parameter will be added to the */etc/exports* file. If the entry already exists in the file, it will be replaced with this new export entry. If the file does not exist, it will be created and the export entry will be added to it. Note that the ″ignore″

flag -I is implied when the ″file″ flag -F is specified without the ″unexport″ flag -U. All export characteristic options provided with the ″options″ flag -O are stored in the */etc/exports* file as given on the command.

-F and -A
This combination is not allowed.

Note: Successful use of the -F flag will cause the contents of the */etc/exports* file to be replaced completely such that it reflects the changes, additions, or deletions caused by the -F flag. Any unrelated existing entries are copied, however ALL comments in the */etc/exports* file will be lost as a result of using the -F flag.

-E This is the ″escape message″ flag and it indicates that an escape message should be issued if the command fails for any of the exports attempted.

-O This flag specifies the export characteristics for the directory tree that is to be exported (made available to NFS clients). The options list following the -O flag list consists of options separated by commas. Some options are followed by an equal ′=′ and a value (or list of values separated by colons ′:′). The options list may contain spaces. If an option is not specified, the default value for that option will be used. The -O flag is only valid when either the ″ignore″ flag -I or the ″file″ flag -F is specified.

If options are required and the -O flag is not specified, the following are the default options.

- ′RW=′ All host names have read-write access to the directory tree.
- ANON=UID associated with the profile *QNFSANON*.
- Requests to bits in the mode other than the permission bits are allowed.
- ′ROOT=′ Root access is not allowed for any hosts.
- ′ACCESS=′ All clients are allowed to mount the directory.
- ′SEC=SYS′ Security is UNIX-like (not recommended for NFS version 4).
- ′VERS=3:2′ NFS version 3 and NFS version 2 are the versions allowed to connect.

The following are the available options and their descriptions.

RO Specifies the protection for the exported directory tree. If RO is specified, the directory tree is exported allowing only read-only access to the directory and its contents. If it is not specified, read-write access is allowed to the directory and its contents.

RW=[HOSTNAME[:HOSTNAME]](...)
Specifies the host name or host names which will be allowed read-write access to the exported directory and its contents. For host names not specified, the directory and its contents will be exported allowing only read-only access.

If neither RO or RW is specified, then ′RW=′ is assumed, and all host names have read-write access to the exported directory.

ANON=UID
If a request comes in from an unknown user, use this UID as the effective userid. Note that root users are considered unknown, unless specified on the ROOT option below. The default value for this option is the UID associated with the user profile *QNFSANON*.

If the user does not want to allow any requests from unknown users, use ′ANON=-1′.

NOSUID

Specifies that any attempt by the client to enable bits other than the permission bits will be ignored. If this option is not specified, attempt to set bits other than the permission bits will be carried out.

ROOT=[HOSTNAME[:HOSTNAME]](...)

Specifies the host name or host names for which root access is allowed to the exported directory tree. If root access is allowed for a host, an incoming UID of 0 is mapped to the user profile QSECOFR, and incoming requests from users with all object (*ALLOBJ) special authority are allowed. If root access is not allowed for a host, an incoming UID of 0 and incoming requests from users with *ALLOBJ special authority are mapped to the UID provided in the ANON option. If the ROOT option is not specified, no hosts will be granted root access.

ACCESS=[CLIENT[:CLIENT]](...)

Specifies the client or clients that are allowed to mount the exported directory tree. A client can be a host name or a netgroup. If no clients are specified, all clients will be allowed to mount the directory tree.

SEC=[SEC[:SEC]](...)

Specifies which security flavors are supported for this particular export entry. Available flavors are:

sys UNIX-like (user ids, group ids).

krb5 Kerberos 5, no integrity or privacy. Only valid when NFS version 4 specified.

krb5i Kerberos 5, with integrity. Only valid when NFS version 4 specified.

krb5p Kerberos 5, with privacy. Only valid when NFS version 4 specified.

VERS=[VERS[:VERS]](...)

Specifies which NFS versions are allowed to mount this export entry. Available versions are 2:3:4. If no version is specified then 3:2 are the default values for this parameter.

EXNAME=[EXPORTED_NAME]

Specifies the name that should be displayed in the client's file system. The export name must be an absolute path name, beginning at the ″root″ (/). Only valid when NFS version 4 specified.

PUBLIC

Specifies that the exported directory is to be the public directory. This must be a subdirectory of the NFSROOT path. The default option is not public. Only valid when NFS version 4 specified.

NFSROOT

NFS version 4 root: Version 4 clients that mount ″/″ will see the specified directory (on DIR parmameter) as the server's root. The default root of the NFS version 4 exported file tree is the system root ″/″. This option can only be used when no other directories are exported. If the NFSROOT path is changed, all future exports must be subdirectories of the new root path. Only valid when NFS version 4 specified.

Top

Directory (DIR)

Specifies the absolute path name of the existing directory to be exported (made available to NFS clients) or unexported (made unavailable to NFS clients). This directory can not be a subdirectory or a parent of an already exported directory (unless it is in a different file system). This parameter is not allowed when the -A flag is specified on the **NFS export options (OPTIONS)** parameter. This parameter is required when the -F flag is specified on the OPTIONS parameter.

Note: A value (other than *NONE) must be specified for either the OPTIONS or DIR parameter. Both OPTIONS and DIR can be specified so long as ′-A′ is not part of the options list specified for the OPTIONS parameter.

Top

Host name (HOSTOPT)

The HOSTOPT parameter has four elements that specify additional information about the NFS clients that a directory tree is to be exported to. If the HOSTOPT parameter is not specified for a host name the user is exporting the directory tree to, the defaults for each of the elements of the HOSTOPT parameter are assumed for that host.

*DFT *DFT specifies that the default values for the elements of the HOSTOPT parameter are used for all clients to which the directory tree or directory trees are to be exported. The network data file coded character set identifier (CCSID) is *BINARY, the network path name CCSID is *ASCII, and Force synchronous write is *SYNC.

Element 1: Host name

The name of the host for which additional options are to be specified. This host should be specified above in the OPTIONS -O list as a host that has access to the exported directory tree. Specify either a single host name that is an alias for an address of a single host or a netgroup name to be associated with these options.

The user can assign names to an internet address with the Work with TCP/IP host table entries option on the Configure TCP/IP menu (CFGTCP) command or via the System i Navigator. Also, a remote name server can be used to map remote system names to internet addresses.

Element 2: Network data file coded character set identifier (CCSID)

The network data file CCSID is used for data of the files sent and received from the specified HOST NAME (or netgroup name). For any hosts not specified on a HOSTOPT parameter, the default network data file CCSID (*BINARY) is used. The CCSID may be one of the following:

*BINARY
The default network data file CCSID (binary, no conversion) is used.

***ASCII**
The ASCII equivalent of the default job CCSID associated with the current job is used.

***JOBCCSID**
The CCSID obtained from the default job CCSID is used.

1-65533
Specify a CCSID for data files.

Element 3: Network path name coded character set identifier (CCSID)

The network path name CCSID is used for the path name components of the files sent to and received from the specified HOST NAME (or netgroup name). For any hosts not specified on a HOSTOPT parameter, the default network path name CCSID (*ASCII) is used. The CCSID may be one of the following:

***ASCII**
The ASCII equivalent of the default job CCSID associated with the current job is used.

***JOBCCSID**
The CCSID obtained from the default job CCSID is used.

1-65533
Specify a CCSID for path name components of files. Only code pages whose CCSIDs can be converted into UCS-2 level 1 (1200) are supported. See Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of supported conversions.

Element 4: Write mode

Specifies whether write requests are handled synchronously or asynchronously for this HOST NAME (or netgroup name). The default value of *SYNC means that data will be written to disk immediately. *ASYNC does not guarantee that data is written to disk immediately, and can be used to improve server performance.

Note: The Network File System (NFS) protocol has traditionally used synchronous writes.

***SYNC**
Write requests are performed synchronously.

***ASYNC**
Write requests are performed asynchronously.

Top

Examples

The alternative command name for EXPORTFS is CHGNFSEXP. The following examples use the alternative command name, but EXPORTFS can be replaced directly for CHGNFSEXP in all of them.

Example 1: Exporting All Entries from /etc/exports

```
CHGNFSEXP   OPTIONS('-A')
 -or-
CHGNFSEXP   '-A'
```

Both of these commands export all entries that exist in the **/etc/exports** file.

Example 2: Exporting One Directory with Options

```
CHGNFSEXP   '-I -O RO,ANON=guest1,ACCESS=Roch1:9.7.431.2'
            '/programs/public'  HOSTOPT((MIAMI1 850 850))
```

This command exports the directory tree under the path name **/programs/public** as read-only. It allows only two clients to mount this directory tree. It takes advantage of the positional parameters OPTIONS and DIR. It uses the HOSTOPT parameter to specify coded character set identifier (CCSID) for the host MIAMI1.

Example 3: Exporting One Directory with Options and Updating the /etc/exports File.

```
CHGNFSEXP   '-I -F -O RO,ANON=guest1,ACCESS=Roch1:9.7.431.2'
            '/programs/public'  HOSTOPT((MIAMI1 850 850))
```

This command exports the directory tree under the path name **/programs/public** as read-only. It allows only two clients to mount this directory tree. The OPTIONS parameter value is specified positionally. It uses the HOSTOPT parameter to specify data and path name coded character set identifiers (CCSIDs) of 850 for host name MIAMI1.

In addition, it also adds an export entry for **/programs/public**, along with the OPTIONS and HOSTOPT parameter values, to the **/etc/exports** file.

Example 4: Exporting One Directory with NFS version 4 Options

```
CHGNFSEXP   '-I -O RO,VERS=4,SEC=KRB5,ACCESS=9.7.431.2'
            '/programs/public'
```

This command exports the directory tree under the path name **/programs/public** as read-only. It allows only one client to mount this directory tree. It allows access only through NFS version 4 protocol and requires a minimum of Kerberos version 5 authentication.

Top

Error messages

***ESCAPE Messages**

CPFA089
Pattern not allowed in path name.

CPFA0CE
Error occurred with path name parameter specified.

CPFA1B8
*IOSYSCFG authority required to use &1.

CPFB41C
&1 entries exported, &2 entries not exported.

Top

Extract Program Information (EXTPGMINF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Extract Program Information (EXTPGMINF) command extracts external linkage information from extended program model (EPM) program objects, and stores this information in a library information file. External linkage information, which includes external variables and entry points, can only be extracted from EPM program objects. The C/400*, FORTRAN/400*, and Pascal compilers produce EPM program objects.

A library information file is a collection of the linkage information for a set of related programs. The library information file name is used on the LIBFILE parameter of the SETPGMINF command. For example, a library information file is provided for the set of programs that make up the C/400 run-time library.

The EXTPGMINF command lets you create a file to store the names of all the affected entry points in your application, instead of specifying each program object name on the SUBPGM parameter of the SETPGMINF command.

Error messages for EXTPGMINF

None

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, **<u>*LIBL</u>**, *USRLIBL, *CURLIB	
FILE	File to receive information	*Qualified object name*	Required, Positional 2
	Qualifier 1: File to receive information	*Name*	
	Qualifier 2: Library	*Name*, **<u>*LIBL</u>**, *CURLIB	
OPTION	Extract record options	**<u>*REPLACE</u>**, *DELETE, *UPDATE	Optional, Positional 3
CRTFILE	Create the file	**<u>*NO</u>**, *YES	Optional
RECLIB	Library name to record	*Name*, **<u>*LIBL</u>**, *FOUND	Optional
CHECK	Consistency Check	**<u>*ALL</u>**, *ENTRY, *DATA, *NONE	Optional

Top

Program (PGM)

The PGM parameter specifies the name of the program and library that contains the linkage information you want to extract. This is a required parameter.

program-name
Enter the name of a program that contains the linkage information you want to extract.

*generic**
Enter a generic name for the programs that contain the linkage information you want to extract.

***ALL** Linkage information is extracted from all the programs that exist in the library.

The possible library values are:

***LIBL** The system searches the library list. You cannot specify *LIBL if you specify CRTFILE (*YES).

***USRLIBL**
The system searches the user portion of the library list.

***CURLIB**
The name of the current library is used. If you have not specified the current library, QGPL is used.

Top

File to receive information (FILE)

Specifies the name and library of the library information file. If the file does not exist, specify CRTFILE(*YES) to create it. If you do not, a message is issued.

file-name
Enter the name of the file where the linkage information will be stored.

The possible library values are:

***LIBL** The system searches the library list.

***CURLIB**
The name of the current library is used. If you have not specified a current library, QGPL is used.

library-name
Enter the name of the library where the linkage information file is located.

Top

Extract record options (OPTION)

Specifies the option of replacing, deleting, or updating data in the library information file.

***REPLACE**
Clears all data in the library information file, and replaces it with the information extracted from the program specified on the PGM parameter.

***DELETE**
Deletes the data in the library information file for the program specified on the PGM parameter. The deletion of data from this library information file results in a compressed file. Data that does not relate to the program specified on the PGM parameter remains in the library information file. No new information is added to the library information file.

***UPDATE**
Deletes the existing information for the specified programs, and replaces it with new information. The deletion of data from this library information file results in a compressed file. If the specified library information file is empty, this option is equivalent to *REPLACE.

Top

Create the file (CRTFILE)

Creates a library information file to store the extracted information.

***NO** Does not create a library information file.

***YES** Creates a library information file to store the extracted information. Select *YES when the specified library information file does not exist. If the file exists, a message is displayed.

Top

Library name to record (RECLIB)

Specifies the name of the library where the programs are stored. At run-time and when you enter the SETPGMINF command, the system searches for programs in the library you specify here.

***LIBL** The system searches the library list.

***FOUND**
During the processing of the EXTPGMINF command, the system records the name of the library where the specified programs are found. The system searches for the library that contained the programs at the time the EXTPGMINF command was processed.

library-name
Enter the name of the library.

Top

Consistency Check (CHECK)

Specifies that the data and entry points in your library information file are checked for consistency. If *NONE is specified, then no warning message will be given; otherwise a PSE warning message will be issued.

***ALL** Checks consistency of both data and entry points.

***ENTRY**
Checks consistency of entry points.

***DATA**
Checks consistency of data.

***NONE**
No consistency checking is performed.

Top

Examples

None

Top

Error messages

None

Top

File Document (FILDOC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The File Document (FILDOC) command allows a user to file a document in the document library.

Restrictions:

- You can file a document on behalf of another user if you are authorized to work on behalf of the other user. You must be granted authority to work on behalf of another with the Grant User Permission (GRTUSRPMN) command.
- The user ID and address must be enrolled in the system distribution directory.
- Security for the new document is taken from the parameters in the FILDOC command and not inherited from the folder.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Information to be filed	*FILE, *IDP, *DSTID	Required, Positional 1
TODOC	To document	*Character value*, ***NONE**	Optional
TOFLR	To folder	*Character value*, ***NONE**	Optional
SENSITIV	Sensitivity	***NONE**, *PERSONAL, *PRIVATE, *CONFIDENTIAL	Optional
USRAUT	User authority	Single values: ***NONE** Other values (up to 50 repetitions): *Element list*	Optional
	Element 1: User profile	*Name*, *PUBLIC	
	Element 2: Authority level	***EXCLUDE**, *USE, *CHANGE, *ALL, *AUTL	
AUTL	Authorization list	*Name*, ***NONE**	Optional
ACC	Access code	Single values: ***NONE** Other values (up to 50 repetitions): 0-2047	Optional
ALWRPL	Allow replacement	***NO**, *YES	Optional
IDPFILE	Profile file	Single values: ***NONE**, *DOCFILE, *DSTIDIDP Other values: *Qualified object name*	Optional
	Qualifier 1: Profile file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
IDPMBR	Profile member	*Name*, ***FIRST**	Optional
USRID	User identifier	Single values: ***CURRENT** Other values: *Element list*	Optional
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
DOCFILE	Document file	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Document file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DOCMBR	Document member	*Name*, ***FIRST**	Optional

Keyword	Description	Choices	Notes
DSTID	Distribution identifier	*Character value*, ***NONE**	Optional
DSTIDEXN	Distribution ID extension	1-99, ***NONE**	Optional
KEEP	Keep in mail log	***NO**, *YES, *REF	Optional
DOCTYPE	Document type	2-65535, ***DFT**, *FFT, *RFT	Optional
SYSCOD	Document system code	*Character value*, ***DFT**	Optional
DOCD	Document description	*Character value*, ***DFT**	Optional
AUTHOR	Author	Single values: ***NONE**, *USRID Other values (up to 50 repetitions): *Character value*	Optional
DOCCLS	Document class	*Character value*, ***NONE**	Optional
KWD	Keyword	Single values: ***NONE** Other values (up to 50 repetitions): *Character value*	Optional
SUBJECT	Subject	Single values: ***NONE**, *DOCD Other values (up to 50 repetitions): *Character value*	Optional
DOCDATE	Document date	*Date*, ***NONE**, *CURRENT	Optional
FILCAB	File cabinet location	*Character value*, ***NONE**	Optional
CPYLST	Copy list	Single values: ***NONE** Other values (up to 50 repetitions): *Character value*	Optional
EXPDATE	Expiration date	*Date*, ***NONE**	Optional
REFERENCE	Reference	*Character value*, ***NONE**	Optional
ACTDATE	Action due date	*Date*, ***NONE**, *CURRENT	Optional
STATUS	Document status	*Character value*, ***NONE**	Optional
CMPDATE	Completion date	*Date*, ***NONE**, *CURRENT	Optional
PROJECT	Project	*Character value*, ***NONE**	Optional
DOCCHRID	Document character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	
DOCLANGID	Language ID	*Character value*, ***JOB**	Optional
DOCCNTRYID	Country or region ID	*Character value*, ***JOB**	Optional
PERSONAL	Personal	***NO**, *YES	Optional
DSTEXPDATE	Distribution expiry indicator	*Element list*	Optional
	Element 1: Date	*Date*, ***NONE**, *CURRENT	
	Element 2: Time	*Time*, ***ENDOFDAY**	
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	

Top

Information to be filed (TYPE)

Specifies the type of information being filed and the parameters that are valid on this command.

This is a required parameter.

***FILE** The database file specified on the **Document file (DOCFILE)** parameter and the **Document member (DOCMBR)** parameter parameter is filed.

Note: If this value is specified, you must specify the default values on the FILCAB, DSTID, DSTIDEXN, and KEEP parameters and you cannot specify DOCFILE(*NONE).

***IDP** The interchange document profile (IDP) specified on the **Profile file (IDPFILE)** parameter and the **Profile member (IDPMBR)** parameter, or the document profile built by this command, is filed.

Note: If this value is specified, you must specify the default values on the DOCFILE, DOCMBR, DOCTYPE, SYSCODE, DOCCHRID, DSTID, DSTIDEXT, and KEEP parameters. If this value is specified, and IDPFILE and FILCAB cannot both specify *NONE.

***DSTID**
The distribution document identified by the distribution identifier specified in the **Distribution identifier (DSTID)** parameter is filed from the mail log into the document library.

Note: If this value is specified, you cannot specify DSTID(*NONE).

Top

To document (TODOC)

Specifies the name of the newly filed document.

name Specify the user-assigned name of the newly filed document. A maximum of 12 characters can be specified. This document name must not exist in the folder that the document is being filed into.

Top

To folder (TOFLR)

Specifies the name of the folder that contains the newly filed document. This parameter can be specified only when a value is also specified on the **To document (TODOC)** parameter.

***NONE**
The newly filed document does not have a user-assigned name and is not filed in a folder.

name Specify the name of the folder to contain the newly filed document. A folder name can consist of a series of folder names (FLR1/FLR2/etc.) if the document is being filed in a folder that is contained in another folder. A maximum of 63 characters can be specified.

You must specify a folder name when a document name is specified on the **To document (TODOC)** parameter.

Top

Sensitivity (SENSITIV)

Specifies the level of sensitivity defined by the X.400 standard. The four levels include no sensitivity, personal, private and company confidential. Any document marked as private is still available to users who are normally authorized to it, but is unavailable to users who are working on your behalf (even though it may be available to them when they are not working on your behalf).

***NONE**
The document has no sensitivity restrictions.

***PERSONAL**
The document is intended for the user as an individual.

***PRIVATE**
The document contains information that should be accessed only by the owner.

***CONFIDENTIAL**
The document contains information that should be handled according to company procedures.

Top

User authority (USRAUT)

Specifies name of an existing user and the user authority level. This parameter is used to change the authorized users for this document by giving more users authority to the document, removing a user's authority for the document, or changing the user's authority to the document.

***NONE**
No users have specific authority to access the document.

***PUBLIC**
Authority is given to the users who do not have specific authority to the document, who are not on the authorization list, and whose user's group does not have specific authority to the document.

name Specify the user profile names of one or more users being given authority to the document.

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

***CHANGE**
The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**
The user cannot access the object.

***AUTL**
The authority of the authorization list specified on the **Authorization list (AUTL)** parameter is used for the document. The *AUTL value is valid only if *PUBLIC is also specified.

Top

Authorization list (AUTL)

Specifies the name of an authorization list used to secure the document specified on the **To document (TODOC)** parameter.

***NONE**
An authorization list is not specified.

name Specify the name of the authorization list whose authority is used for the document.

Top

Access code (ACC)

Specifies the access codes used with this document. The access codes must already exist on the system. If they do not already exist, they must be added on the system with the Add Access Code (ADDACC) command.

***NONE**
No access codes are assigned to this document. Authority for this document is controlled by the values specified on the **User authority (USRAUT)** parameter and the **Authorization list (AUTL)** parameter.

0-2047 Specify the access codes that control who can use the document. A maximum of 50 access codes can be specified. Access code 0 gives use (*USE) authority to all users.

Top

Allow replacement (ALWRPL)

Specifies the setting to allow replacement of the document content of the document being filed. If this parameter is specified when filing a document that cannot be replaced, it is ignored. A document that cannot be replaced cannot be changed back to a document that can be replaced.

***NO** The document content of the document being filed cannot be replaced.

***YES** The document content of the document being filed can be replaced.

Top

Profile file (IDPFILE)

Specifies where the document profile information is located. If you specify this parameter, the remaining parameters after the **Profile member (IDPMBR)** parameter are ignored, except the **Command character identifier (CMDCHRID)** parameter and the **Document character identifier (DOCCHRID)** parameter.

***NONE**
The interchange document profile (IDP) is supplied by other parameters on this command. There is no database file containing the IDP information. If *NONE is specified, the **Profile member (IDPMBR)** parameter is ignored.

***DSTIDIDP**
The IDP information associated with the distribution document is used. The **Profile member (IDPMBR)** parameter is ignored. This is valid only when TYPE (*DSTID) is specified.

***DOCFILE**
The database file specified for the document also contains the profile information. If *DOCFILE is specified, the **Document file (DOCFILE)** parameter and **Document member (DOCMBR)** parameter are used for the document profile information.

name Specify the name of the database file that contains the IDP. The document profile database file can be a user-defined file or the output file specified on the Receive Distribution (RCVDST) or Retrieve Document (RTVDOC) commands. If you specify a user-defined file, it must have the same format as the output file produced by RCVDST or RTVDOC. If an output file is specified, only the data portion of the document profile record is read from the output file. The prefix is removed from the document profile record.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

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

name Specify the library where the database file is located.

Top

Profile member (IDPMBR)

Specifies the interchange document file member name being used. This parameter is used only when a database file name is also specified on the **Profile file (IDPFILE)** parameter.

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

name Specify the name of the database file member being used.

Top

User identifier (USRID)

Specifies which user ID and user ID address should be associated with the request.

Single values

***CURRENT**
You are performing the request for yourself.

Element 1: User ID

character
Specify another user's user ID or your user ID. You must have been given permission to work on behalf of another user or have all object (*ALLOBJ) special authority.

Element 2: Address

character
Specify another user's address or your address. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

Top

Document file (DOCFILE)

Specifies the names of the database file and the library that contains the document data. The database file is a user-defined file or the output file specified in either the Receive Distribution (RCVDST) command or the Retrieve Document (RTVDOC) command. If an output file is specified, only the data portion of the document data record is read from the output file. The prefix is removed from the document data record.

name Specify the name of the database file that contains the document data.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

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

name Specify the library where the database file is located.

Top

Document member (DOCMBR)

Specifies the document database file member that is used.

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

name Specify the name of the database file member that is used.

Top

Distribution identifier (DSTID)

Specifies the unique distribution identifier of the distribution. The distribution identifier is assigned by the system when the distribution is originated. Distribution identifiers can be found by using the Query Distribution (QRYDST) command. Identifiers are also returned from the Send Distribution (SNDDST) command.

***NONE**
No distribution identifier is used.

distribution-id
Specify the 3-part distribution identifier which is composed of the second part of the sender's user ID (padded on the right to 8 characters), the first part of the sender's user ID (padded on the right to 8 characters), and a 4-digit zoned sequence number with leading zeros. For example, 'NEWYORK SMITH 0204'. This parameter is required when *DSTID is specified on the **Information to be sent (TYPE)** parameter.

Top

Distribution ID extension (DSTIDEXN)

Specifies the extension of the distribution identifier (if any) specified on the **Distribution identifier (DSTID)** parameter. This 2-digit extension has a value ranging from 01 through 99 that uniquely identifies duplicate distributions. The default value is 01.

***NONE**
There is no duplicate distribution. *NONE is equivalent to an extension of 01.

distribution-id-extension
Specify the extension associated with the distribution. This is used to uniquely identify duplicate distributions.

Top

Keep in mail log (KEEP)

Specifies whether to keep a copy of the distribution document filed in the mail log, delete the distribution from the mail log, or keep a reference in the mail log of the filed distribution document.

***NO** Delete the distribution document from the mail log after the file is complete.

***YES** Keep a copy of the filed distribution document in the mail log.

***REF** The distribution document is deleted and a reference to the filed distribution document is kept in the mail log.

Top

Document type (DOCTYPE)

Specifies the type of document being used. This identifier is used by the system to determine whether the data stream can be handled properly.

***DFT** The system creates the proper document type identifier based on the source of the data.

***FFT** The document is in Final Form Text. This type of document is intended to be viewed and printed, but not edited, by the receiver.

***RFT** The document is in Revisable Form Text. This type of document can be viewed, printed, and edited by the receiver.

2-65,535
Specify a document type identifier value. The numbers from 2 through 32,767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM-defined document types. The numbers ranging from 32,768 through 65,535 are not registered with IBM and can be used for non-IBM-defined document types. The meaning of these document types must be determined by defining the value of the system code on the **System code (SYSCOD)** parameter.

Top

System code (SYSCOD)

Specifies the text used with the value specified on the **Document type (DOCTYPE)** parameter to help uniquely identify the type of document being used. The receiver of the data stream determines the document data stream and processing requirements to edit, view, print, or change the document.

***DFT** The system supplies a default system code. If the value specified on the **Document type (DOCTYPE)** parameter is a number ranging from 2 through 32,767, the default is retrieved from message CPX9026. If the value specified on the **Document type (DOCTYPE)** parameter is in the range from 32,768 through 65,535, a system code must be specified.

system-code
Specify the text that uniquely identifies the type of document being sent. A maximum of 13 characters can be specified.

Top

Document description (DOCD)

Specifies a description for the document being filed. This description is in the Document Interchange Architecture document name field.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

***DFT** The system creates a document description from the database files. The default is (library-name/file-name/member-name) for database files. If *IDP is specified on the **Information to be filed (TYPE)** parameter to file only a reference to a printed document, the default document name is Hardcopy Document Reference and is retrieved from the message CPX9025. An installation may change this message, but only the first 44 characters are used in the document name. If *DSTID is specified on the **Information to be filed (TYPE)** parameter, the default document name will be the distribution document name specified when the distribution was sent.

description
Specify the description of the document. A maximum of 44 characters can be specified.

Top

Author (AUTHOR)

Specifies the author or authors of the document.

***NONE**
No author is identified for the document.

***USRID**
The user ID and address specified on the **User identifier (USRID)** parameter is used as the author's name.

name Specify the name of the author or authors. A maximum of 50 authors can be specified.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Document class (DOCCLS)

Specifies the class associated with this document, such as MEMO, FORM, or SHEET.

***NONE**
No class is assigned to the document.

class Specify the document class. A maximum of 16 characters can be specified.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Keyword (KWD)

Specifies the keywords that describe the document.

***NONE**
No keywords are defined for this document.

keyword
Specify the keywords to describe the document. A maximum of 50 keywords can be specified. Each keyword can have a maximum of 60 characters.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Subject (SUBJECT)

Specifies the subject or subjects of the document.

***NONE**
No subject is defined for the document.

***DOCD**
The document description is used as the subject for the document.

subject
Specify the subject or subjects of the document. A maximum of 50 subjects can be specified and each subject can have a maximum of 60 characters of text.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Document date (DOCDATE)

Specifies any date the user needs to assign to the document.

***NONE**
No date is assigned to the document.

***CURRENT**
The system assigns the current system date to the document.

date Specify the document date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

File cabinet location (FILCAB)

Specifies the location of the document. This parameter is intended to describe the location of printed documents. The interchange document profile (IDP) that refers to the printed document is distributed. This parameter is required if *IDP is also specified on the **Information to be sent (TYPE)** parameter and *NONE is specified on the **Profile file (IDPFILE)** parameter.

***NONE**
No filing cabinet reference is defined for this document.

reference
Specify the text that describes where the printed document is located. A maximum of 60 characters can be specified.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Copy list (CPYLST)

Specifies the names of the users that receive this document.

***NONE**
No copy list is included for this document.

name Specify the names of the users that receive the document. A maximum of 50 names can be specified. Each name can have a maximum of 60 characters.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Expiration date (EXPDATE)

Specifies the date on which the document is no longer needed.

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

date Specify the document expiration date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Reference (REFERENCE)

Specifies a reference associated with the document.

***NONE**
No reference field is included for this document distribution.

reference
Specify text that describes the reference associated with the document. A maximum of 60 characters can be used.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Action due date (ACTDATE)

Specifies the date when the action requested is due.

***NONE**
No action due date request is specified.

***CURRENT**
The current system action due date is used.

date Specify the action due date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Document status (STATUS)

Specifies the user-defined status of the document. Examples of status are: In Process, Pending Approval, or Retired.

***NONE**
No status is included in this document.

status Specify text that describes the status of the document. A maximum of 20 characters can be specified.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Completion date (CMPDATE)

Specifies the date when the requested action is completed.

***NONE**
No completion date is included.

***CURRENT**
The current system date is used as the completion date.

date Specify the action completion date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Project (PROJECT)

Specifies the project associated with the document.

***NONE**
No project field information is included in this document.

name Specify the name of the project associated with the document. A maximum of 10 characters can be specified.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Document character identifier (DOCCHRID)

Specifies the character identifier (graphic character set and code page) for the document data being used. The character identifier is related to the display device used to create the document data.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

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

***DEVD**
The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

graphic-character-set code-page
Specify the graphic character set and code page values used to create the data being distributed.

Note: Both parts can be up to 5 digits in length.

Top

Language ID (DOCLANGID)

Specifies the language identifier to be placed in this document's interchange document profile (IDP).

***JOB** The language identifier specified for the job in which this command is entered is used.

language-identifier
Specify a language identifier. Press the PF4 key from the **Language ID (DOCLANGID)** parameter to see a list of valid identifiers.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Country or region ID (DOCCNTRYID)

Specifies the country or region identifier to be placed in this document's interchange document profile (IDP).

***JOB** The country or region identifier specified for the job in which this command is entered is used.

country-or-region-ID
Specify a country or region identifier. Press the PF4 key from the **Country or region ID (DOCCNTRYID)** parameter to see a list of valid identifiers.

Note: This parameter is ignored if the **Profile file (IDPFILE)** parameter is specified.

Top

Personal (PERSONAL)

Specifies whether the document distribution is private or not. This parameter is replaced by SENSITIV but the PERSONAL parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the SENSITIV parameter.

If PERSONAL(*YES) is used, the SENSITIV parameter must be omitted or it must be with the value SENSITIV(*NONE). If the command is prompted without this parameter specified, this parameter is not displayed.

***NO** Only the owner and users that have authorization to the document can get access to documents that are not sensitive. Users authorized to work on behalf of other users who have access to the document can access documents that are not sensitive. This value will map to SENSITIV(*NONE).

***YES** Only the owner can get access to private documents. Users authorized to work on behalf of other users who have access to the document cannot get access to the document. This value will map to SENSITIV(*PRIVATE).

Top

Distribution expiry indicator (DSTEXPDATE)

Specifies the date and time on which the distribution is no longer needed in the mail log.

***NONE**
No expiration date,

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

date Specify the value to use as the expiration date. The date must be specified in the format specified by the system value QDATFMT.

Element 2: Time

***ENDOFDAY**
An expiration time is requested by the end of the specified date. The time is set to 23:59:59.

time Specify the value used as the expiration time.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

The CMDCHRID parameter applies to the following parameters and means that the character set and code page are stored with the fields to allow the display station that accesses the document to correctly print or display the fields. The fields are translated to a common character set and code page when the fields are written to the search database. The interchangeable character set and code page are '697 500'.

The following fields are translated:

- **User identifier** (USRID)
- **Distribution identifier** (DSTID)
- **Document system code** (SYSCOD)
- **Document description** (DOCD)
- **Author** (AUTHOR)
- **Document class** (DOCCLS)
- **Keyword** (KWD)
- **Subject** (SUBJECT)
- **File cabinet location** (FILCAB)
- **Copy list** (CPYLST)
- **Reference** (REFERENCE)
- **Document status** (STATUS)
- **Project** (PROJECT)

Single values

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

***DEVD**
The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767
Specify the graphic character set to use.

Element 2: Code page

1-32767
Specify the code page to use.

Top

Examples

Example 1: Filing a Personal Document

```
FILDOC   TYPE(*FILE)  DOCFILE(MARYLIB/MARYFILE)
         SENSITIV(*PRIVATE)  IDPFILE(*DOCFILE)
```

This command files a private document using a database file that has the document content and document profile information. The default for the distribution ID extension is 01 (DSTID(01)).

Example 2: Filing a Distribution Document

```
FILDOC   TYPE(*DSTID)  DSTID('NEWYORK SMITH 0201')  DSTID(02)
         DOCCLS('NEW CLASS')  TODOC(DST0201)  TOFLR(FLRDST)
```

This command files a distribution document in the document library QDOC in document DST0201 and folder FLRDST. The document class is changed in the distribution document, and the second distribution that was sent to the user is filed.

Top

Error messages

***ESCAPE Messages**

CPF900B
User ID and address &1 &2 not in System Distribution Directory.

CPF900C
Sign on and verify of user failed.

CPF901B
Document filing request failed.

CPF902B
Authority of *AUTL is allowed only with USRAUT(*PUBLIC).

CPF905C
Error occurred trying to find a translation table.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9845
Error occurred while opening file &1.

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

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

Top

Format Data (FMTDTA)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Format Data (FMTDTA) command processes a series of Sort specifications stored in a source file member.

Top

Parameters

Keyword	Description	Choices	Notes
INFILE	Input file	Values (up to 8 repetitions): *Element list*	Required, Positional 1
	Element 1: File	*Qualified object name*	
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**, *LIBL	
	Element 2: Member	*Name*, ***FIRST**	
OUTFILE	Output file	*Element list*	Required, Positional 2
	Element 1: File	*Qualified object name*	
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**, *LIBL	
	Element 2: Member	*Name*, ***FIRST**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 3
	Qualifier 1: Source file	*Name*, **QFMTSRC**	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
SRCMBR	Source member	*Name*, ***FIRST**	Optional, Positional 4
PRTFILE	Print file	*Qualified object name*	Optional, Positional 5
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTION	Options:	Values (up to 8 repetitions): *CHK, *NOCHK, *PRT, *NOPRT, *DUMP, *NODUMP, *NOSECLVL, *SECLVL	Optional, Positional 6
PGMDATE	Program date:	*Date*, ***CURRENT**	Optional

Top

Input file (INFILE)

Specifies up to eight names for files that are to be used as input. For database files, one member name can be specified for each file name. For diskette files, the diskette identifier can be specified for each device file name. This is a required parameter.

file-name
Enter the name of the file that is to be used as input.

***CURLIB**
The current library will be used. If you have not specified a current library, QGPL will be used.

***LIBL** The system searches the library list to find the library where the file is located.

library-name
Enter the name of the library of the input file.

***FIRST**
The first member in the file is to be used as input.

data-file-identifier
For diskette files, enter one data file identifier per device file name specified. If more than one diskette data file is to be processed for a device file name, the device file name should be specified as many times as required.

member-name
For database files, enter one member name per database file name specified. If more than one member of the same database file is to be processed, the database file name should be specified as many times as required.

Top

Output file (OUTFILE)

Specifies the name of the file and the name of the member to be used for output. Both the file and member must exist before being named in this parameter. This is a required parameter.

file-name
Enter the name of the output file to be used.

***CURLIB**
The current library will be used. If you have not specified a current library, QGPL will be used.

***LIBL** The system searches the library list to find the library where the file is located.

library-name
Enter the name of the library of the output file.

***FIRST**
The first member in the file is to be used for output.

member-name
Enter the name of the member in the output file that is to be used for output.

Top

Source file (SRCFILE)

Specifies the name of the source file containing the sort specifications to be run. The source file may be a device or database file, and it must have the attributes of a source file.

QFMTSRC
The IBM-supplied source file QFMTSRC contains the sort specifications.

source-file-name
Enter the name of the source file that contains the sort specifications.

***LIBL** The system searches the library list to find the library where the source file is located.

***CURLIB**
The current library will be used. If you have not specified a current library, QGPL will be used.

library-name
Enter the name of the library that contains the source file.

Top

Source member (SRCMBR)

Specifies the name of the source file member containing the sort specifications to be run. The source file may be a device or database file, and it must have the attributes of a source file.

***FIRST**
The first member of the source file containing the sort specifications is to be run.

data-file-identifier
Enter the name of the diskette data file identifier that contains the sort specification statements, if the data file resides on diskette.

member-name
Enter the name of the member of the source file containing the sort specifications to be run.

Top

Print file (PRTFILE)

Specifies the name of the printer device file to which the print data is to be sent.

QSYSPRT
The data is to be printed by the system printer.

print-file-name
Enter the name of the printer device file that is to print the data.

***LIBL** The system searches the library list to find the library where the file is located.

***CURLIB**
The current library will be used. If you have not specified a current library, QGPL will be used.

library-name
Enter the name of the library that contains the file.

Top

Options: (OPTION)

Specifies the sequence checking and printing options to be used while the sort utility is running.

***CHK** The sort specifications are to be sequence-checked.

***NOCHK**
The sort specifications are not to be sequence-checked.

***PRT** The sort specifications and any error or informational messages are to be printed.

***NOPRT**
The sort specifications and any error or informational messages are not to be printed.

***NODUMP**
The internal tables used for problem analysis are not to be printed.

***DUMP**
The internal tables used for problem analysis are to be printed.

***NOSECLVL**
Suppresses the printing of second level text for errors detected during compilation.

***SECLVL**
Prints second level text for errors detected during compilation.

Top

Program date: (PGMDATE)

Specifies the date that can be used with factor 2 as a keyword in record specifications.

***CURRENT**
Use the current system date when the command is processed.

***DATE**
Enter the date in the format specified by system value QDHTFMT, or if separators are used, by QDATSEP.

Top

Examples

None

Top

Error messages

Unknown

Top

Generate Message Catalog (GENCAT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

This command is an alias for the Merge Message Catalog (MRGMSGCLG) command and can also be issued using the following alternative command name:

- MRGMSGCLG

The Generate Message Catalog (GENCAT) command merges message text from one or more source files (SRCFILE parameter) with message text in the specified message catalog (CLGFILE parameter). If the catalog specified does not already exist, it will be created using values specified for the CLGCCSID, DTAAUT, and OBJAUT parameters. If the catalog already exists, the CCSID, DTAAUT, and OBJAUT attributes of the existing message catalog will be used.

You can specify up to 300 message text source files. Message text source files are processed in the sequence specified. Each successive source file modifies the catalog. If a message number in the source file already exists in the message catalog, the new message text defined in the source file replaces the old message text in the message catalog file. If a message number in the source file does not already exist in the message catalog, the message information is added to the message catalog.

Top

Parameters

Keyword	Description	Choices	Notes
CLGFILE	Message catalog name	*Path name*	Required, Positional 1
SRCFILE	Source file path name	Values (up to 300 repetitions): *Path name*	Required, Positional 2
TEXT	Text ’description’	*Character value*, ***BLANK**	Optional
CLGCCSID	Message catalog CCSID	1-65533, ***SRCCCSID**, *JOB	Optional
SRCCCSID	Source file CCSID	1-65533, ***SRCFILE**, *JOB	Optional
DTAAUT	Public authority for data	*Name*, ***INDIR**, *NONE, *RWX, *RX, *RW, *WX, *R, *W, *X, *EXCLUDE	Optional
OBJAUT	Public authority for object	Single values: ***INDIR**, *NONE, *ALL Other values (up to 4 repetitions): *OBJEXIST, *OBJMGT, *OBJALTER, *OBJREF	Optional

Top

Message catalog name (CLGFILE)

Specifies the path name of the message catalog to be changed or created. All directories in a stream file path name must exist. If no stream file exists with the specified path name, a message catalog with the specified file name is created. If the path name is in the QSYS file system, the file must exist. If a file member in the QSYS file system does not exist, it is created. Source physical files with multiple data fields are not supported.

Top

Source file path name (SRCFILE)

Specifies the path name of the source file that contains the message text to be merged into the message catalog. If the file is from the QSYS file system, then it must be a database source physical file.

Note: If the source file is not a record file, then each line in the source file must have been terminated with a newline or linefeed character when the source file was created.

Top

Text ’description’ (TEXT)

Specifies the text that briefly describes the message catalog.

Note: Assigning text to objects is dependent on the support provided by the file system or object type used for the message catalog.

The possible values are:

***BLANK**
The mode name consisting of 8 blank characters is used.

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

Top

Message catalog CCSID (CLGCCSID)

Specifies the coded character set ID (CCSID) in which to store the message text in the message catalog. If the message catalog is a stream file, the CCSID value entered is used to set the stream file’s attributes. Use the Work with Object Links (WRKLNK) command to display the CCSID of a message catalog. Use the Display File Description (DSPFD) command to determine the CCSID of a message catalog in the QSYS file system.

The possible values are:

***SRCCCSID**
Special value indicating that the CCSID will be determined from the value specified for the source file CCSID (SRCCSID parameter).

***JOB** Special value indicating the job CCSID is used for the catalog information. If the job CCSID is 65535, the job default CCSID is used.

coded-character-set-ID
Specify the CCSID used for the catalog information. The values 0, 65534, and 65535 are not valid.

Top

Source file CCSID (SRCCCSID)

Specifies the coded character set ID (CCSID) of the source file.

The possible values are:

***SRCFILE**
Special value indicating that the CCSID will be determined from the CCSID of the first source file (SRCFILE parameter).

***JOB** Special value indicating the job CCSID is used for the CCSID of the source file. If the job CCSID is 65535, the job default CCSID is used.

coded-character-set-ID
Specify the CCSID of the source file. The values 0, 65534, and 65535 are not valid.

Top

Public authority for data (DTAAUT)

Specifies the public authority given users for the data in the object created.

The possible values are:

***INDIR**
The authority for the object being created is determined by the directory it is being created in. If *INDIR is used for DTAAUT, it is also required for OBJAUT.

***RWX** The users are given *RWX authority to the objects. *RWX authority allows the user to perform all operations on the object except those limited to the owner or controlled by object existence, object management, object alter, and object reference authority. The user can change the object and perform basic functions on the object. *RWX authority provides object operational authority and all the data authorities.

***RX** *RX authority allows the user to perform basic operations on the object, such as run a program or display the contents of a file. The user is prevented from changing the object. *RX authority provides object operational authority and read and execute authorities.

***RW** *RW authority allows the user to view the contents of an object and modify the contents of an object. *RW authority provides object operational authority and data read, add, update, and delete authorities.

***WX** *WX authority allows the user to modify the contents of an object and run a program or search a library or directory. *WX authority provides object operational authority and data add, update, delete, and execute authorities.

***R** *R authority allows the user to view the contents of an object. *R authority provides object operational authority and data read authority.

***W** *W authority allows the user to modify the contents of an object. *W authority provides object operational authority and data add, update, and delete authorities.

***X** *X authority allows the user to run a program or search a library or directory. *X authority provides object operational authority and data execute authority.

***EXCLUDE**
Exclude authority prevents the user from accessing the object. The OBJAUT value must be *NONE if this special value is used.

***NONE**
The users will not be given any of the data authorities to the objects. This value cannot be used with OBJAUT value of *NONE.

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

Top

Public authority for object (OBJAUT)

Specifies the authorities given users to the object.

The possible values are:

***INDIR**
The object authority is based on the authority for the directory where this object is being created. If *INDIR is used for DTAAUT, it is also required for OBJAUT.

***NONE**
None of the other object authorities (existence, management, alter, or reference) will be given to the users. If *EXCLUDE or an authorization list name is specified for the DTAAUT parameter, this value must be specified.

***ALL** All of the other object authorities (existence, management, alter, and reference) will be given to the users.

Or specify up to four (4) of the following values:

***OBJEXIST**
The users will be given object existence authority to the object.

***OBJMGT**
The users will be given object management authority to the object.

***OBJALTER**
The users will be given object alter authority to the object.

***OBJREF**
The users will be given object reference authority to the object.

Top

Examples

Examples for MRGMSGCLG

```
MRGMSGCLG   CLGFILE('/USDIR/USMSG.CAT')  CLGCCSID(*SRCCSID)
            SRCFILE('/QSYS.LIB/MYLIB.LIB/MSGSRC.FILE/USMSG.MBR')
            DTAAUT(*R)  TEXT('Message catalog for USA')
```

This command merges the message text from member USMSG of source physical file MSGSRC in library MYLIB in the QSYS file system with message catalog USMSG.CAT in directory USDIR. If the message catalog does not already exist, it will be created with the CCSID of the source file and data authority of *R. The text parameter describes this as a message catalog for the USA.

Top

Error messages

***ESCAPE Messages**

CPF3BE3
Message catalog &1 not created or updated.

Top

Generate Keystore File Entry (GENCKMKSFE)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Generate Keystore File Entry (GENCKMKSFE) command generates a random key or key pair and stores it in a keystore file.

For more information on keystore, refer to the Cryptographic Services Keystore article in the Cryptographic Services section of the APIs topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

- You must have object operational (*OBJOPR), read (*READ) and add (*ADD) authorities to the keystore file.

Top

Parameters

Keyword	Description	Choices	Notes
KEYSTORE	Keystore file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Keystore file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RCDLBL	Record label	*Character value*	Required, Positional 2
KEYTYPE	Key type	*MD5, *SHA1, *SHA256, *SHA384, *SHA512, *DES, *TDES, *AES, *RC2, *RC4, *RSA	Required, Positional 3
KEYSIZE	Key size	1-4096	Required, Positional 4
EXPONENT	Public key exponent	3, **65537**	Optional
DISALLOW	Disallowed function	Values (up to 3 repetitions): ***NONE**, *ENCRYPT, *DECRYPT, *MAC, *SIGN	Optional

Top

Keystore file (KEYSTORE)

Specifies the keystore file to use.

This is a required parameter.

Qualifier 1: Keystore file

name Specify the name of the keystore file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

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

name Specify the name of the library to search for the file.

Top

Record label (RCDLBL)

Specifies the label of a key record in the specified keystore file.

This is a required parameter.

character-value
Specify the key record label. The label can be up to 32 characters and contain any alphanumeric characters.

Top

Key type (KEYTYPE)

Specifies the algorithm type of the key.

This is a required parameter.

***MD5** An MD5 key is used for hash message authentication code (HMAC) operations. Because of weaknesses in the algorithm, MD5 should not be used except for compatibility purposes. The minimum length for an MD5 HMAC key is 16 bytes. A key longer than 16 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 64 bytes will be hashed before it is used.

***SHA1**
An SHA-1 key is used for HMAC operations. Because of weaknesses in the algorithm, SHA-1 should not be used except for compatibility purposes. The minimum length for an SHA-1 HMAC key is 20 bytes. A key longer than 20 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 64 bytes will be hashed before it is used.

***SHA256**
An SHA-256 key is used for HMAC operations. The minimum length for an SHA-256 HMAC key is 32 bytes. A key longer than 32 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 64 bytes will be hashed before it is used.

***SHA384**
An SHA-384 key is used for HMAC operations. The minimum length for an SHA-384 HMAC key is 48 bytes. A key longer than 48 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 128 bytes will be hashed before it is used.

***SHA512**
An SHA-512 key is used for HMAC operations. The minimum length for an SHA-512 HMAC key is 64 bytes. A key longer than 64 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 128 bytes will be hashed before it is used.

***DES** An older, widely used symmetric encryption algorithm. DES should not be used except for compatibility purposes. Only 7 bits of each byte are used as the actual key. The rightmost bit of each byte will be set to odd parity because some cryptographic service providers require that a DES key have odd parity in every byte. The key size parameter must specify 8.

***TDES**
A symmetric encryption algorithm that improves the security of DES by performing the DES algorithm three times. Only 7 bits of each byte are used as the actual key. The rightmost bit of each byte will be set to odd parity because some cryptographic service providers require that a DES key have odd parity in every byte. The key size can be 8, 16, or 24. Triple DES operates on an encryption block by doing a DES encrypt, followed by a DES decrypt, and then another DES encrypt. Therefore, it actually uses three 8-byte DES keys. If the key is 24 bytes in length, the first 8 bytes are used for key 1, the second 8 bytes for key 2, and the third 8 bytes for key 3. If the key is 16 bytes in length, the first 8 bytes are used for key 1 and key 3, and the second 8 bytes for key 2. If the key is only 8 bytes in length, it will be used for all 3 keys (essentially making the operation equivalent to a single DES operation).

***AES** A newly developed symmetric encryption algorithm designed to replace DES. AES offers faster and stronger encryption than TDES. The key size can be 16, 24, or 32.

***RC2** A variable-key-size symmetric encryption algorithm. The key size can be 1 - 128.

***RC4** A variable-key-size symmetric stream encryption algorithm. The key size can be 1 - 256. Because of the nature of the RC4 operation, using the same key for more than one message will severely compromise security.

***RSA** An asymmetric encryption algorithm that uses a public/private key pair. The key size is the modulus length, specified in bits, and must be an even number in the range 512 - 4096. Both the RSA public and private key parts are stored in the key record.

Top

Key size (KEYSIZE)

Specifies the size of key to generate. For RSA keys this length is specified in bits. For all other keys it is specified in bytes. Refer to the key type parameter for restrictions.

This is a required parameter.

unsigned-integer
Specify the size of the key to generate.

Top

Public key exponent (EXPONENT)

Specifies the public-key exponent for an RSA key pair. To maximize performance, the public-key exponent is limited to the following two values. The value of 65537 may be more secure than a value of 3.

Note: This parameter will be ignored if any value other than *RSA is specified for the **Key type (KEYTYPE)** parameter.

65537

3

Top

Disallowed function (DISALLOW)

Specifies the functions that cannot be used with this key record. Multiple functions can be disallowed.

Single values

***NONE**
This key is allowed to be used in all cryptographic functions.

Other values (up to 3 repetitions)

***ENCRYPT**
This key is not allowed to be used in encryption operations.

***DECRYPT**
This key is not allowed to be used in decryption operations.

***MAC** This key is not allowed to be used in message authentication code (MAC) operations.

***SIGN**
This key is not allowed to be used in digital signing operations.

Top

Examples

Example 1: Generate an AES Keystore Entry

```
GENCKMKSFE   KEYSTORE(MYLIB/MYKEYSTORE) RCDLBL('Byllesby')
             TYPE(*AES) SIZE(32)
```

This command generates a 32-byte (256-bit) AES key and stores it in keystore file MYKEYSTORE in library MYLIB.

Example 2: Generate an RSA Keystore Entry

```
GENCKMKSFE   KEYSTORE(MYLIB/MYKEYSTORE) RCDLBL('Pepin')
             TYPE(*RSA) SIZE(2048) EXPONENT(3)
             DISALLOW(*ENCRYPT *DECRYPT *MAC)
```

This command generates a 2048-bit RSA public/private key pair that can only be used in digital signing and verification operations.

Top

Error messages

***ESCAPE Messages**

CPF3CF2
Error(s) occurred during running of &1 API.

CPF9872
Program or service program &1 in library &2 ended. Reason code &3.

CPF9D94
A pending value exists for a master key.

CPF9D9E
Record label already exists.

CPF9D9F
User not authorized to key store file.

CPF9DA0
Error opening key store file.

CPF9DA5
Key store file not found.

CPF9DA6
Key store file is not available.

CPF9DA7
File is corrupt or not a valid key store file.

CPF9DB3
Qualified keystore file name is not valid.

CPF9DB6
Record label not valid.

CPF9DB7
Error occured writing to the key store file.

CPF9DB8
Error occured reading record from key store.

CPF9DDA
Unexpected return code &1 from cryptographic service provider &2.

Top

Generate Command Documentation (GENCMDDOC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Generate Command Documentation (GENCMDDOC) command generates an output file which contains documentation for a Control Language (CL) command. The generated file will be one of the following:

- If *HTML is specified for the **Generation options (GENOPT)** parameter, the file will contain HyperText Markup Language (HTML) source. The generated file can be displayed using a standard internet browser, and conforms to HTML 4.0 specifications. The information used to generate the file is retrieved from the specified command (*CMD) object and any command help panel group (*PNLGRP) objects associated with the command.
- If *UIM is specified for the GENOPT parameter, the file will contain User Interface Manager (UIM) source. The generated source is an outline for the online command help for the command. The information used to generate the file is retrieved only from the specified command (*CMD) object. This option is intended to simplify the task of writing online help for CL commands.

  See the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/, for more information on writing command documentation using UIM.

Restrictions:

- You must have use (*USE) authority to the specified command and execute (*EXECUTE) authority for the library that the command is in. If a generic name or *ALL is specified for the command name, no output file is generated for any commands that you do not have *USE authority for.
- For each associated panel group that contains command help information for the specified command, you must have *USE authority to the panel group and *EXECUTE authority for the library that the panel group is in.
- You must have execute (*X) authority to the directories in the path for the generated file, and write and execute (*WX) authorities to the parent directory of the generated file.
- If the output file does not exist, the public authority will be determined by the **os400.file.create.auth** Java property value. If this Java property has not been set, the public authority for a created stream file is set to *RW.
- If the output file already exists, you must have write (*W) authority to the file and *YES must be specified for the **Replace file (REPLACE)** parameter.
- This command does not support proxy CL commands.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	*Qualified object name*	Required, Positional 1
	Qualifier 1: Command	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TODIR	To directory	*Path name*, **'.'**	Optional
TOSTMF	To stream file	*Character value*, ***CMD**	Optional

Keyword	Description	Choices	Notes
REPLACE	Replace file	***YES**, *NO	Optional
GENOPT	Generation options	Values (up to 3 repetitions): *HTML, *UIM, *NOSHOWCHOICEPGMVAL, *SHOWCHOICEPGMVAL, *NOSERVICE, *SERVICE	Optional

Top

Command (CMD)

Specifies the command for which a documentation output file is to be generated.

Note: If a generic command name or *ALL is specified for the command name, *LIBL is not allowed as the library name qualifier and the value for the **To stream file (TOSTMF)** parameter must be *CMD.

This is a required parameter.

Qualifier 1: Command

***ALL** Documentation files for all of the commands in the specified library are to be generated.

generic-name
Specify the generic name of the commands for which documentation files are to be generated. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all commands that have names with the same prefix as the generic command name will have documentation files generated.

name Specify the name of the command for which you want to generate a documentation output file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

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

name Specify the name of the library where the command is located.

Top

To directory (TODIR)

Specifies the directory in which the generated command documentation file will be stored. The file name to be used within this directory is specified by the **To stream file (TOSTMF)** parameter.

'.' The output file will be stored in the current working directory.

path-name
Specify the path name for the directory where you want the generated output file stored.

Top

To stream file (TOSTMF)

Specifies the target stream file to be used to store the generated command documentation file. The specified file will be located using the directory path specified for the **To directory (TODIR)** parameter.

Note: If a generic command name or *ALL is specified for the **Command (CMD)** parameter, the value specified or defaulted for this parameter must be *CMD.

***CMD** If the TODIR parameter specifies that the target is in the /QSYS.LIB file system, the generated file name will be same as the command name.

Otherwise, the generated file name depends on whether *HTML or *UIM is specified for the **Generation options (GENOPT)** parameter. If *HTML is specified, the generated file name will be **libname_cmdname.html**, where **cmdname** is the command name and **libname** is the name of the library where the command is located. If *UIM is specified, the generated file name will be **libname_cmdname.uim**

character-value
Specify the name to be used for the generated command documentation file.

Top

Replace file (REPLACE)

Specifies whether or not to replace an existing file in the target directory (TODIR parameter) by the file name specified or generated (TOSTMF parameter).

***YES** If a file already exists by the name specified or implied, the file contents will be replaced with the generated command documentation file.

***NO** If a file already exists by the name specified or implied, an error message is sent and no command documentation file is generated. If no file by the same name exists in the target directory, the file will be created and no error message sent.

Top

Generation options (GENOPT)

Specifies options to control the command information to be generated. Multiple option values can be specified in any order on this parameter. If neither or both of the values in each group are specified, the underlined value will be used.

Note: The underlined values for this parameter are *similar* to, but not *actually* default values, and therefore, cannot be changed with the Change Command Default (CHGCMDDFT) command.

Generated Source Option

***HTML**
The generated file will contain HyperText Markup Language (HTML) source. The generated file can be displayed using a standard internet browser, and conforms to HTML 4.0 specifications. The information used to generate the file is retrieved from the specified command (*CMD) object and any command help panel group (*PNLGRP) objects associated with the command.

***UIM** The generated file will contain User Interface Manager (UIM) source. The generated source is an outline for the online command help for the command. The information used to generate the file is retrieved only from the specified command (*CMD) object. This option is intended to simplify the task of writing online help for CL commands. After editing the generated file to add descriptive text for the command and storing the source in a source file member, this UIM source can be used as input to the Create Panel Group (CRTPNLGRP) command to create a command help panel group for the command.

See the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/, for more information on writing command documentation using UIM.

Choice Program Values Option

***NOSHOWCHOICEPGMVAL**
For command parameters that have an associated *choices program*, do not show the values which would be returned from the choices program in the generated parameter summary table. Choices program values can vary from system to system. Not showing the choices program values gives you just the parameter values defined in the command object.

***SHOWCHOICEPGMVAL**
For command parameters that have an associated *choices program*, show the values returned from calling the choices program in the generated parameter summary table. Showing the choice program values gives you the same parameter values that you would see if prompting the command on this system.

Service Option

***NOSERVICE**
No extra trace or dump information is generated.

***SERVICE**
This option is intended to be used if the command is not working and you are told by your software service provider to write an APAR for the problem. Specifying this option will cause additional trace and dump information to be generated. Send this additional generated information with the APAR.

Top

Examples

Example 1: Generating HTML Documentation for an i5/OS Command

```
GENCMDDOC   CMD(CRTUSRPRF)
```

This command generates a documentation file for the CRTUSRPRF command. The command will be located using the library list for the current thread. The generated stream file will be stored in the current working directory of the job. Assuming the command is found in library QSYS, the generated file name will be **QSYS_CRTUSRPRF.html**. If a file by that name already exists in the target directory, it will be replaced by the generated file.

Example 2: Generating UIM Documentation for User Command

```
GENCMDDOC   CMD(MYLIB/MYCMD)
            TODIR('/QSYS.LIB/MYLIB.LIB/QPNLSRC.FILE')
            TOSTMF('MYCMD.MBR')  REPLACE(*NO)  GENOPT(*UIM)
```

This command generates a documentation file for command MYCMD which is located in library MYLIB. The generated file will be stored in file QPNLSRC in library MYLIB with a member name of MYCMD. If a member already exists in the target file with this name, an error message will be sent and no documentation file will be generated.

Top

Error messages

***ESCAPE Messages**

CPF6E74
&1 command documents failed; &2 command documents created successfully.

CPF6E75
Error detected on the CMD parameter.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9899
Error occurred during processing of command.

CPFA09C
Not authorized to object. Object is &1.

CPFA0A0
Object already exists. Object is &1.

CPFA0A9
Object not found. Object is &1.

CPF6E67
Command documentation not generated for proxy command &1 in &2.

Top

Generate JVM Dump (GENJVMDMP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Generate JVM Dump (GENJVMDMP) command will generate Java Virtual Machine (JVM) dumps upon request.

Restrictions:

The issuer of the command must be running under a user profile which is the same as the job user identity of the job being worked with, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority.

The job user identity is the name of the user profile by which a job is known to other jobs. More information about the job user identity is in the Work management topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	*Qualified job name*	Required, Positional 1
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
TYPE	Type	Values (up to 3 repetitions): ***JAVA**, *SYSTEM, *HEAP	Optional

Top

Job name (JOB)

Specifies the name of the job where the JVM is running. All three of the qualifiers for the job parameter are required.

This is a required parameter.

Qualifier 1: Job

name Specify the name of the job for which to generate a dump.

Qualifier 2: User

name Specify the name that identifies the user profile under which the job was started.

Qualifier 3: Number

000000-999999
Specify the job number assigned by the system.

Top

Type (TYPE)

Specifies the type of dump to generate.

You can specify 3 values for this parameter.

***JAVA** Generates multiple files that contain diagnostic information for the JVM and the Java applications running within the JVM.

***SYSTEM**
Generate a binary format raw memory image of the job that was running when the dump was initiated.

***HEAP**
Generates a dump of all the heap space allocations which have not yet been freed.

Top

Examples

Example 1: Generate Javadump

```
GENJVMDMP  JOB(032072/USERNAME/QP0ZSPWP) TYPE(*JAVA)
```

This command will cause a Javadump for the Java Virtual Machine running the job 032072/USERNAME/QP0ZSPWP.

Example 2: Generate Three Java VM Dumps

```
GENJVMDMP  JOB(032072/USERNAME/QP0ZSPWP) TYPE(*JAVA *SYSTEM *HEAP)
```

This command will generate a Java core dump, system dump, and heap dump for the Java virtual machine running the job 032072/USERNAME/QP0ZSPWP.

Top

Error messages

***ESCAPE Messages**

CPE3440
Operation not supported.

CPF9871
Error occurred while processing.

JVAB321
Java Virtual Machine job &3/&2/&1 not found or not active.

JVAB322
Not authorized to perform function on Java Virtual Machine.

JVAB323
Operation not supported with a Classic Java Virtual Machine.

JVAB328
Machine or ASP storage limit reached.

JVAB332
GENJVMDMP failed.

Top

Go to Menu (GO)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The Go to Menu (GO) command shows the menu requested. This command allows you to specify either a particular menu or a generic menu name. You can optionally specify whether or not to return to the menu from which the command is entered after showing the menu specified.

Using the Previous and Exit Keys

A menu is placed on an internal menu stack before it is run. If a stack is not available for the menu, one is created. When the Cancel key is pressed for a menu, the previous menu in the stack is shown. Each menu stack is ten elements (menus) deep. When the eleventh menu is placed on the menu stack, the first, or oldest, menu is removed from the stack. This menu cannot be returned to by using the Cancel key.

Pressing the Exit key returns the user to the last display or menu from which a GO command was entered with RTNPNT(*YES). The display that the user is returned to is found by removing menus from the menu stack until a return point is found. This process may also cause a program in the call stack to return to its calling program unless the program is a return point.

Pressing either the Exit or Cancel key while viewing help for a menu returns the user to the menu.

Restrictions:

- You must have use (*USE) authority for the menu and its display and message files or program (whichever applies).
- You must also have *USE authority for the library where the menu is located.

Top

Parameters

Keyword	Description	Choices	Notes
MENU	Menu	*Qualified object name*	Required, Positional 1
	Qualifier 1: Menu	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL, *ALL, *ALLUSR	
RTNPNT	Return point	***YES**, *NO	Optional, Positional 2

Top

Menu (MENU)

Specifies the menu to be shown.

This is a required parameter.

Qualifier 1: Menu

***ALL** A list of all menus in the specified library is shown from which you select the menu to be run.

generic-name
Specify the generic name of the menu to be run. A generic name is a character string that contains one or more characters followed by an asterisk (*). A list of all menus that have names that begin with the same characters as the generic menu name is shown from which you select the menu to be run.

name Specify the name of the menu to be shown.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

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

***USRLIBL**
If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

***ALLUSR**
All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB     #DSULIB     #SEULIB
#COBLIB     #RPGLIB
#DFULIB     #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

```
QDSNX       QRCLxxxxx   QUSRDIRDB   QUSRVI
QGPL        QSRVAGT     QUSRIJS     QUSRVxRxMx
QGPL38      QSYS2       QUSRINFSKR
QMGTC       QSYS2xxxxx  QUSRNOTES
QMGTC2      QS36F       QUSROND
QMPGDATA    QUSER38     QUSRPOSGS
QMQMDATA    QUSRADSM    QUSRPOSSA
QMQMPROC    QUSRBRM     QUSRPYMSVR
QPFRDATA    QUSRDIRCF   QUSRRDARS
QRCL        QUSRDIRCL   QUSRSYS
```

1. ’xxxxx’ is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***ALL** All libraries in the system, including QSYS, are searched.

name Specify the name of the library to be searched.

Top

Return point (RTNPNT)

Specifies whether to return to the display where the command is entered when the Exit key is pressed.

***YES** The display where the command is entered is returned to when the Exit key is pressed.

***NO** The display where the command is entered is not returned to when the Exit key is pressed.

Top

Examples

```
GO   MENU(PERSMENU)
```

This command runs a menu called PERSMENU, located in a library found by searching the library list (*LIBL default value).

If the Exit key is pressed while PERSMENU is being shown, the display where the GO command was entered (*YES default value on the RTNPNT parameter) is shown.

Top

Error messages

***ESCAPE Messages**

CPF6ACD
Menu &1 in &2 is wrong version for system.

CPF6AC7
Menu &1 in library &2 not displayed.

Top

Go To (GOTO)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The Go To (GOTO) command is used in a CL program or ILE CL procedure for branching from one part of the program to another. The branching is to the label on another command that is specified on the Go To (GOTO) command. Branching can be either forward or backward, but the specified label must be inside the program. For a GOTO command in a subroutine, the branch target must be associated with the same subroutine.

Restrictions:

- This command is valid only within a CL program or ILE CL procedure.
- Using GOTO to branch into or out of a subroutine is not allowed.

Top

Parameters

Keyword	Description	Choices	Notes
CMDLBL	Command label	*Simple name*	Required, Positional 1

Top

Command label (CMDLBL)

Specifies the label name of the command to which control is transferred; when the Go To (GOTO) command is processed. The command with the label is then processed. If the specified command cannot be run (for example, if it is a DCL command), control is transferred to the next command following the command with the specified label. The label must be within the same program as the GOTO command. A CL variable name cannot be used to specify the label name.

This is a required parameter.

Top

Examples

```
LOOP:  CHGVAR &A (&A + 1)
IF  (&A *LT 30)  THEN(GOTO  LOOP)
```

The Change Variable (CHGVAR) command increases the value of &A by 1 until &A is equal to or greater than 30. The GOTO command is processed each time that the IF command tests the expression and the result is true; the GOTO command following the THEN parameter causes the procedure to branch back to the label LOOP on the CHGVAR command. Refer to the descriptions of the CHGVAR command and the IF command for additional explanations of their functions.

Top

Error messages

None

Top

Grant Access Code Authority (GRTACCAUT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Grant Access Code Authority (GRTACCAUT) command allows you to give the specified users authority to access documents and folders associated with the access codes. Access is restricted to read only (*USE authority).

Restrictions:

- The access code must be defined to the system before you can grant access code authority.
- The user being granted access code authority must be enrolled in the system distribution directory.
- To use this command, you must have all object (*ALLOBJ) or security administrator (*SECADM) special authority.

Top

Parameters

Keyword	Description	Choices	Notes
ACC	Document access code	Single values: *REFUSER Other values (up to 300 repetitions): 1-2047	Required, Positional 1
USER	User profile	Values (up to 300 repetitions): *Name*	Required, Positional 2
REFUSER	Reference user profile	*Name*, ***NONE**	Optional

Top

Document access code (ACC)

Specifies the access code being authorized for use by the user identified on the **User profile (USER)** parameter.

This is a required parameter.

Single values

***REFUSER**
The access code authority being granted is based on the (referred to) user profile name; specified for the **Reference user profile (REFUSER)** parameter.

Other values (up to 300 repetitions)

1-2047 Specify the access code to which you want authority to be granted. The access code must be defined to the system using the Add Access Code (ADDACC) command before being specified on this parameter.

Top

User profile (USER)

Specifies the user profile name of the users to whom you are granting access code authority. The users identified will have the access code added to their current list of authorized access codes; this access code is used to verify additional document and folder accesses from the document library. The user must be enrolled in the system distribution directory before being granted authority to use an access code.

Note: By granting an access code to a group user profile, that access code is granted (implicitly) to every user under that group.

This is a required parameter.

You can specify 300 values for this parameter.

name Specify the name of the user profile of a user to whom you are granting access code authority.

Top

Reference user profile (REFUSER)

Specifies the referred-to user profile on which the access code authority is based. If this parameter is used, then *REFUSER must be specified for the **Document access code (ACC)** parameter.

*NONE
No referred to user is used to grant access code authority.

name Specify the name of the user profile that the access code authority is based on. This user must also be enrolled in the system distribution directory.

Top

Examples

Example 1: Granting Authority to Multiple Users

```
GRTACCAUT   ACC(3 30 60)  USER(SAM LARRY)
```

This command gives authority to access codes 3, 30, and 60 to SAM and LARRY.

Example 2: Granting Authority Based on Another User

```
GRTACCAUT   ACC(*REFUSER)  USER(JOE)  REFUSER(TOM)
```

This command grants access code authority to JOE based on TOM's authority. For example, if JOE currently has authority to access codes 1, 12, and 50, and TOM currently has authority to access codes 8 and 9, the GRTACCAUT command authorizes JOE to access codes 1, 8, 9, 12, and 50.

Top

Error messages

***ESCAPE Messages**

CPF9009
System requires file &1 in &2 be journaled.

CPF9013
Access code authority given to &1 users, not granted to &2 users.

CPF9024
System cannot get correct record to finish operation.

CPF9065
Not allowed to give access code authority.

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.

Top

Grant Object Authority (GRTOBJAUT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Grant Object Authority (GRTOBJAUT) command grants specific authority for the objects named in the command to another user or group of users.

Authority can be given to:

- Named users
- Users (*PUBLIC) who do not have specific authority to the object or the authorization list
- Users of the object referred to by the **Reference object (REFOBJ)** and **Reference object type (REFOBJTYPE)** and parameters
- Authorization lists

If AUT(*AUTL) is specified, the PUBLIC authority for the object comes from the PUBLIC authority of the authorization list securing the object.

The AUTL parameter is used to secure an object with an authorization list or remove an authorization list from an object. User profiles cannot be secured by an authorization list (*AUTL).

This command can be used by an object's owner, or by a user with object management authority for the specified object. A user with object management authority can grant to other users any authority that the user has, except object management authority. Only the owner of the object, or someone with all object special authority (*ALLOBJ), can grant object management authority to a user.

A user with *ALL authority can assign a new authorization list.

When granting authority to users, the REPLACE parameter indicates whether the authorities you specify replace the user's existing authorities. The default value of REPLACE(*NO) gives the authority that you specify, but it does not remove any authority that is greater than you specified, unless you are granting *EXCLUDE authority. REPLACE(*YES) removes the user's current authorities, then grants the authority that you specify.

When granting authority with a reference object, this command gives the authority that you specify, but it does not remove any authority that is greater than you specified, unless you are granting *EXCLUDE authority.

This command gives the authority that you specify, but it does not remove any authority that is greater than you specified, unless you are granting *EXCLUDE authority or specify REPLACE(*YES).

Restrictions:

1. This command must get an exclusive lock on a database file before read or object operational authority can be given to a user.
2. If a user requests authority for another specified user to a device currently in use by another authorized user, authority to the device is not given.
3. Object type *AUTL cannot be specified.
4. AUT(*AUTL) is valid only with USER(*PUBLIC).
5. A user must either be the owner of the object or have *ALL authority to use the AUTL parameter.
6. The user must have object management authority to the object.

7. If the object is a file, the user must have object operational and object management authorities.
8. For display stations or for work station message queues associated with the display station, if this command is not entered at the device for which authorities are to be granted, it should be preceded by the Allocate Object (ALCOBJ) command and followed by the Deallocate Object (DLCOBJ) command.
9. You must have *USE authority to the auxiliary storage pool device if one is specified.

Note: Caution should be used when changing the public authority on IBM-supplied objects. For example, changing the public authority on the QSYSOPR message queue to be more restrictive than *CHANGE will cause some system programs to fail. The system programs will not have enough authority to send messages to the QSYSOPR message queue. For more information, refer to the System i Security Reference, SC41-5302 book.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Generic name*, *name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *ALL, *ALLUSR, *USRLIBL, *ALLAVL, *ALLUSRAVL	
OBJTYPE	Object type	*ALL, *ALRTBL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSD, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *S36, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
ASPDEV	ASP device	*Name*, *, *SYSBAS	Optional
USER	Users	Single values: *PUBLIC Other values (up to 50 repetitions): *Name*	Optional, Positional 3
AUT	Authority	Single values: ***CHANGE**, *ALL, *USE, *EXCLUDE, *AUTL Other values (up to 10 repetitions): *OBJALTER, *OBJEXIST, *OBJMGT, *OBJOPR, *OBJREF, *ADD, *DLT, *READ, *UPD, *EXECUTE	Optional, Positional 4
AUTL	Authorization list	*Name*, *NONE	Optional
REFOBJ	Reference object	*Qualified object name*	Optional
	Qualifier 1: Reference object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Keyword	Description	Choices	Notes
REFOBJTYPE	Reference object type	***OBJTYPE**, *ALRTBL, *BNDDIR, *AUTL, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSD, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *S36, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Optional
REFASPDEV	Reference ASP device	*Name*, *****, *SYSBAS	Optional
REPLACE	Replace authority	***NO**, *YES	Optional

Top

Object (OBJ)

Specifies the objects for which specific authority is to be given to one or more users.

This is a required parameter.

Qualifier 1: Object

***ALL** Specific authority is to be given to all objects of the specified object type (OBJTYPE parameter). A specific library name must be specified for the library qualifier when *ALL is specified.

generic-name
Specify the generic name of the objects for which specific authority is to be given to one or more users. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all objects that have names with the same prefix as the generic name are shown.

name Specify the name of the object for which specific authority is to be given to one or more users.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***USRLIBL**
If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

***ALLUSR**

All user libraries in the auxiliary storage pools (ASPs) defined by the **ASP device (ASPDEV)** parameter are searched.

User libraries are all libraries with names that do not begin with the letter Q except for the following:

```
#CGULIB
#COBLIB
#DFULIB
#DSULIB
#RPGLIB
#SDALIB
#SEULIB
```

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

```
QDSNX
QGPL
QGPL38
QMGTC
QMGTC2
QMPGDATA
QMQMDATA
QMQMPROC
QPFRDATA
QRCL
QRCLxxxxx
QSRVAGT
QSYS2
QSYS2xxxxx
QS36F
QUSER38
QUSRADSM
QUSRBRM
QUSRDIRCF
QUSRDIRCL
QUSRDIRDB
QUSRIJS
QUSRINFSKR
QUSRNOTES
QUSROND
QUSRPOSGS
QUSRPOSSA
QUSRPYMSVR
QUSRRDARS
QUSRSYS
QUSRVI
QUSRVxRxMx
```

1. ’xxxxx’ is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***ALLAVL**

All libraries in all available ASPs are searched.

***ALLUSRAVL**

All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

name Specify the name of the library to be searched.

Top

Object type (OBJTYPE)

Specifies the object type of the object for which specific authorities are to be given to the specified users or to an authorization list. For a complete list of supported object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt).

This is a required parameter.

***ALL** Specific authorities for all supported object types are given to the specified users or to the authorization list.

object-type

Specify the object type of the object for which specific authorities are to be given to the specified users.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device name where the library that contains the object (OBJ parameter) is located. If the object's library resides in an ASP that is not part of the library name space associated with the job, this parameter must be specified to ensure the correct object is used as the target of this command's operation.

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

***SYSBAS**
The system ASP and all basic user ASPs will be searched to locate the object. No independent ASPs will be searched, even if the job has an ASP group.

name
Specify the device name of the independent ASP to be searched to locate the object. The independent ASP must have been activated (by varying on the ASP device) and have a status of AVAILABLE. The system ASP and basic user ASPs will not be searched.

Top

Users (USER)

Specifies one or more users to whom authority for the named object is to be given.

This is a required parameter unless either the **Reference object (REFOBJ)** parameter or **Authorization list (AUTL)** parameter is specified.

***PUBLIC**
Users are authorized to use the object as specified in the AUT parameter when they do not have authority specifically given to them for the object, are not on the authorization list and none of their groups have any authority or are on the authorization list. Users who do not have any authority, and whose groups do not have any authority, are authorized to use the object as specified in the AUT parameter.

name
Specify the names of one or more users to be given specific authority for the object. Up to 50 user profile names can be specified.

Top

Authority (AUT)

Specifies the authority to be given to the users specified for the **Users (USER)** parameter.

If a value is specified for this parameter, you cannot specify a value for the AUTL, REFOBJ, or REFOBJTYPE parameters.

Single values

***CHANGE**
The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

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

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**
The user cannot access the workstation object.

***AUTL**
The public authority of the authorization list specified on the AUTL parameter is used for the public authority for the object.

Note: You can specify AUT(*AUTL) only when USER(*PUBLIC) is also specified.

Other values (up to 10 repetitions)

***OBJALTER**
Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJMGT**
Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

***OBJEXIST**
Object existence authority provides the authority to control the object's existence and ownership. If a user has special save system authority (*SAVSYS), object existence authority is not needed to perform save restore operations on the object.

***OBJOPR**
Object operational authority provides authority to look at the description of an object and use the object as determined by the data authority that the user has to the object.

***OBJREF**
Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

Data authorities

***ADD** Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

***DLT** Delete authority provides the authority to remove entries from an object.

***EXECUTE**
Execute authority provides the authority needed to run a program or locate an object in a library.

***READ**
Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

***UPD** Update authority provides the authority to change the entries in an object.

Top

Authorization list (AUTL)

Specifies the authorization list whose entries are to be used to grant authority for the object specified. You must have authorization list management (*AUTLMGT) authority for the specified authorization list.

If a value is specified for this parameter, you cannot specify a value for the AUT, REFOBJ, or REFOBJTYPE parameters.

***NONE**
The authorization list that secures the object is removed. If public authority in the object is *AUTL, it is changed to *EXCLUDE.

name Specify the name of the authorization list to be used.

Top

Reference object (REFOBJ)

Specifies the reference object to be queried to obtain authorization information. Those authorizations are given to the object specified by the OBJ and OBJTYPE parameters. Users authorized to the reference object are authorized in the same manner to the object for which authority is to be given. If the reference object is secured by an authorization list, that authorization list secures the object specified by the OBJ and OBJTYPE parameters.

If a value is specified for this parameter, you cannot specify a value for the AUT or AUTL parameters.

name Specify the name of the reference object.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

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

name Specify the name of the library to be searched.

Top

Reference object type (REFOBJTYPE)

Specifies the object type of the reference object specified for the **Reference object (REFOBJ)** parameter.

***OBJTYPE**
The object type of the reference object is the same as the object type specified for the **Object type (OBJTYPE)** parameter.

object-type
Specify the object type of the reference object. To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt).

Top

Reference ASP device (REFASPDEV)

Specifies the auxiliary storage pool (ASP) device name where the library that contains the reference object (REFOBJ parameter) is located. If the reference object's library resides in an ASP that is not part of the library name space associated with the job, this parameter must be specified to ensure the correct object is queried for authorities.

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

***SYSBAS**
The system ASP and all basic user ASPs will be searched to locate the reference object. No independent ASPs will be searched, even if the job has an ASP group.

name Specify the device name of the independent ASP to be searched to locate the reference object. The independent ASP must have been activated (by varying on the ASP device) and have a status of AVAILABLE. The system ASP and basic user ASPs will not be searched.

Top

Replace authority (REPLACE)

Specifies whether the authorities replace the user's current authorities.

***NO** The authorities are given to the user, but no authorities are removed, unless you are granting *EXCLUDE authority.

***YES** The user's current authorities are removed, then the authorities are given to the user.

Top

Examples

Example 1: Granting Authority to All Users

```
GRTOBJAUT   OBJ(USERLIB/PROGRAM1)  OBJTYPE(*PGM)  USER(*PUBLIC)
```

This command gives authority to use the object named PROGRAM1 to all users of the system who do not have authorities specifically given to them, who are not on an authorization list, whose user groups do not have authority to the object, or whose user groups are not on the authorization list. The object is a program (*PGM) located in the library named USERLIB. Because the AUT parameter is not specified, the authority given to all users is change authority. This allows all users to run the program and to debug it.

Example 2: Granting Object Management Authority

```
GRTOBJAUT   OBJ(ARLIB/PROGRAM2)  OBJTYPE(*PGM)  USER(TMSMITH)
            AUT(*OBJMGT)
```

This command gives object management authority to user named TMSMITH. This authority allows TMSMITH to grant to others personally possessed authorities for the object named PROGRAM2, which is a program located in the library named ARLIB.

Example 3: Granting Authority to Users on Authorization List

```
GRTOBJAUT   OBJ(MYLIB/PRGM3)  OBJTYPE(*PGM)  AUTL(KLIST)
```

This command gives to users the authority specified for them on authorization list KLIST for the object named PRGM3. The object is a program located in library MYLIB.

Top

Error messages

***ESCAPE Messages**

CPF22A0
Authority of *AUTL is allowed only with USER(*PUBLIC).

CPF22A1
OBJTYPE(*AUTL) not valid on this command.

CPF22A2
Authority of *AUTL not allowed for object type *USRPRF.

CPF22A3
AUTL parameter not allowed for object type *USRPRF.

CPF22A9
Authority of *AUTL cannot be specified.

CPF22DA
Operation on file &1 in &2 not allowed.

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF2208
Object &1 in library &3 type *&2 not found.

CPF2209
Library &1 not found.

CPF2210
Operation not allowed for object type *&1.

CPF2211
Not able to allocate object &1 in &3 type *&2.

CPF2216
Not authorized to use library &1.

CPF2223
Not authorized to give authority to object &1 in &3 type *&2.

CPF2227
One or more errors occurred during processing of command.

CPF2236
AUT input value not supported.

CPF2243
Library name &1 not allowed with OBJ(generic name) or OBJ(*ALL).

CPF2245
Process profile not owner of object &1 in &3 type *&2.

CPF2253
No objects found for &1 in library &2.

CPF2254
No libraries found for &1 request.

CPF2273
Authority may not have been changed for object &1 in &3 type *&2 for user &4.

CPF2283
Authorization list &1 does not exist.

CPF2290
*EXCLUDE cannot be specified with another authority.

CPF9804
Object &2 in library &3 damaged.

Top

Grant User Authority (GRTUSRAUT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Grant User Authority (GRTUSRAUT) command grants authority to a user by referring to another user profile.

Note: You should use group support or authorization lists instead of the Grant User Authority (GRTUSRAUT) command support whenever possible for better performance in granting authority and the subsequent SAVSYS or SAVSECDTA function.

If a security officer issues this command, the authorities in the user profile are granted to the receiving user, including object management authority.

If this command is run by the owner of the user profile, all authorities for each object owned are granted, including object management authority.

For objects that the user profile being referred to does not own but is authorized to use, the user of this command must have object management authority and the authorities to be granted for the object, or must own the object. Otherwise, no authority is granted for the object.

Ownership of objects or authorities held by a user profile cannot be changed by this command. Authorities to objects granted to a user profile are added to any authorities that the user profile already had.

Restrictions:

- The following user profiles cannot be specified for either of the parameters on this command:

  QANZAGENT, QAUTPROF, QCLUMGT, QCLUSTER, QCOLSRV, QDBSHR, QDBSHRDO, QDFTOWN, QDIRSRV, QDLFM, QDOC, QDSNX, QEJB, QEJBSVR, QGATE, QIBMHELP, QIPP, QLPAUTO, QLPINSTALL, QMGTC, QMSF, QNETSPLF, QNFSANON, QNTP, QPEX, QPM400, QRJE, QSNADS, QSPL, QSRVAGT, QSYS, QTCM, QTCP, QTMHHTP1, QTMHHTTP, QTSTRQS, QYCMCIMOM, QYPSJSVR

Top

Parameters

Keyword	Description	Choices	Notes
USER	User	*Name*	Required, Positional 1
REFUSER	Referenced user	*Name*	Required, Positional 2

Top

User (USER)

Specifies the user profile to whom authority is to be granted.

This is a required parameter.

name Specify the name of the user profile.

Top

Referenced user (REFUSER)

Specifies the user profile to be referred to for authority.

This is a required parameter.

name Specify the name of the user profile.

Top

Examples

Example 1: Running GRTUSRAUT under QSECOFR User Profile

```
GRTUSRAUT   USER(USRB)  REFUSER(USRA)
```

This command grants the user profile USRB the same authorities that USRA has for all objects that USRA owns (including object management authority) or has authority to.

Example 2: Running GRTUSRAUT under User Profile USRA

```
GRTUSRAUT   USER(USRB)  REFUSER(USRC)
```

This command grants the user profile USRB the same authorities that USRC has for all objects that USRC has authorities to only if USRA, entering this command, has object management authority to the objects or is the owner of the objects being referred to.

Top

Error messages

***ESCAPE Messages**

CPF2204
User profile &1 not found.

CPF2211
Not able to allocate object &1 in &3 type *&2.

CPF2213
Not able to allocate user profile &1.

CPF2217
Not authorized to user profile &1.

CPF2222
Storage limit is greater than specified for user profile &1.

CPF2223
Not authorized to give authority to object &1 in &3 type *&2.

CPF2252
Authority given to &2 objects. Authority not given to &3 objects.

Top

Grant User Permission (GRTUSRPMN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Grant User Permission (GRTUSRPMN) command allows you to grant permission for a user to handle documents and folders or to perform other tasks on behalf of another user. Access is restricted to documents, folders, and mail items that are not personal. The users specified must be enrolled in the system distribution directory before you run this command.

Restrictions:

The user must have all object (*ALLOBJ) special authority to grant permission for a user to work on behalf of another user.

Top

Parameters

Keyword	Description	Choices	Notes
TOUSER	To user profile	*Name*	Required, Positional 1
FORUSER	For user profile	Single values: ***CURRENT** Other values (up to 300 repetitions): *Name*	Optional, Positional 2

Top

To user profile (TOUSER)

Specifies the name of the user profile that is permitted to work on behalf of the user specified on the **For user profile (FORUSER)** parameter. Access is restricted to documents, folders, and mail items that are not personal. The user profile must exist, and the user must be enrolled in the system distribution directory before you run this command.

Top

For user profile (FORUSER)

Specifies the names of the user profiles for which the user specified on the **To user profile (TOUSER)** parameter works. The users must be enrolled in the system distribution directory before you run this command.

***CURRENT**
You are granting permission to someone working on your behalf.

name Specify the name of the user profile on whose behalf the user specified on the **To user profile (TOUSER)** parameter works.

Top

Examples

```
GRTUSRPMN   TOUSER(JUDY)  FORUSER(PEGGY)
```

JUDY is the administrative assistant for an executive. This command allows JUDY to work with documents or folders for PEGGY that are not personal.

Top

Error messages

***ESCAPE Messages**

CPF9007
User permission given for &1 users, not given &2 users.

CPF9009
System requires file &1 in &2 be journaled.

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.

Top

Grant Workstation Object Aut (GRTWSOAUT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Grant Workstation Object Authority (GRTWSOAUT) command is used by one user to grant specific authority for the workstation object named in this command to another user or group of users. Workstation objects are used by the i5/OS Graphical Operations program.

Authority can be given to:

- Named users.
- Users (*PUBLIC) who do not have authority specifically given to them either for the object or for the authorization list.
- Groups of users who do not have any authority to the object or are not on the authorization list that secures the object.
- Users of the referenced workstation object (specified on the REFWSO parameter).
- Users on an established authorization list.

When AUT(*AUTL) is specified, the user can The authority for:

- All users who do not have authority specifically given to them for an object.
- Users who are not on the authorization list that secures the object.
- Users whose user group does not have authority specifically given to it.
- Users whose user group is not on the authorization list that secures the object.

This command can be used by an object owner, by the security officer, or by a user with object management authority for the specified object.

Restrictions:

1. A user must be either the owner of the object or have *ALL authority to use the AUTL parameter.
2. The user must have object management authority to the object to grant authority to the object.
3. AUT(*AUTL) can be specified only with USER(*PUBLIC). User profile names cannot be secured by an authorization list (*AUTL).
4. Only the owner of the object, or someone with all object authority (*ALLOBJ), can grant object management authority to a user.

Top

Parameters

Keyword	Description	Choices	Notes
WSOTYPE	Workstation object type	*Element list*	Required, Positional 1
	Element 1:	*TPLWRKARA, *WRKARA, *TPLPRTOL, *PRTOL, *TPLPRTL, *PRTL, *TPLOUTQ, *TPLOUTQL, *OUTQL, *TPLJOBL, *JOBL, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSGL, *MSGL, *TPLMSGQ, *TPLMSGSND, *MSGSND, *TPLSGNUSL, *SGNUSL, *TPLOBJL, *OBJL, *TPLLIBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLAUNCH, *PRSSET	

Keyword	Description	Choices	Notes
USER	Users	Single values: *PUBLIC Other values (up to 50 repetitions): *Qualifier list*	Optional, Positional 2
	Qualifier 1: Users	*Name*	
AUT	Authority	Single values: ***CHANGE**, *ALL, *USE, *EXCLUDE, *AUTL Other values (up to 7 repetitions): *OBJEXIST, *OBJMGT, *OBJOPR, *ADD, *DLT, *READ, *UPD	Optional, Positional 3
AUTL	Authorization list	*Name*	Optional
REFWSO	Reference workstation object	*Element list*	Optional
	Element 1:	*TPLWRKARA, *WRKARA, *TPLPRTOL, *PRTOL, *TPLPRTL, *PRTL, *TPLOUTQ, *TPLOUTQL, *OUTQL, *TPLJOBL, *JOBL, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSGL, *MSGL, *TPLMSGQ, *TPLMSGSND, *MSGSND, *TPLSGNUSL, *SGNUSL, *TPLOBJL, *OBJL, *TPLLIBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLAUNCH, *PRSSET	

Top

Workstation object type (WSOTYPE)

Specifies the workstation objects whose authority is to be editted.

This is a required parameter.

***TPLWRKARA**
The work area template is the workstation object.

***WRKARA**
The work area objects are the workstation objects.

***TPLPRTOL**
The printer output list template is the workstation object.

***PRTOL**
The printer output list objects are the workstation objects.

***TPLPRTL**
The printer list template is the workstation object.

***PRTL** The printer list objects are the workstation objects.

***TPLOUTQ**
The output queue template is the workstation object.

***TPLOUTQL**
The output queue list template is the workstation object.

***OUTQL**
The output queue list objects are the workstation objects.

***TPLJOBL**
The job list template is the workstation object.

***JOBL** The job list objects are the workstation objects.

***TPLJOBQ**
The job queue template is the workstation object.

***TPLJOBLOG**
The job log template is the workstation object.

***JOBLOG**
The job log objects are the workstation objects.

***TPLJOBQL**
The job queue list template is the workstation object.

***JOBQL**
The job queue list objects are the workstation objects.

***TPLMSGL**
The message list template is the workstation object.

***MSGL**
The message list objects are the workstation objects.

***TPLMSGQ**
The message queue template is the workstation object.

***TPLMSGSND**
The message sender template is the workstation object.

***MSGSND**
The message sender objects are the workstation objects.

***TPLSGNUSL**
The signed-on user list template is the workstation object.

***SGNUSL**
The signed-on user list objects are the workstation objects.

***TPLOBJL**
The object list template is the workstation object.

***OBJL** The object list objects are the workstation objects.

***TPLLIBSL**
The library list template is the workstation object.

***LIBSL**
The library list objects are the workstation objects.

***TPLLIB**
The library template is the workstation object.

***TPLLAUNCH**
The job submitter template is the workstation object.

***LAUNCH**
The job submitter objects are the workstation objects.

***PRSSET**
The personal settings objects are the workstation objects.

Top

Users (USER)

Specifies one or more users to whom authorities for the named object are to be given. If user names are specified, the authorities are given specifically to those users. Authority given by this command can be revoked specifically by the Revoke Workstation Object Authority (RVKWSOAUT) command.

This is a required parameter unless either the **Reference workstation object (REFWSO)** parameter or **Authorization list (AUTL)** parameter is specified.

***PUBLIC**
All users of the system, who do not have authority specifically given to them for the object, who are not on the authorization list, whose user group does not have any authority, or whose user group is not on the authorization list, are authorized to use the object as specified on the AUT parameter.

name Specify the name of one or more user profiles. A maximum of 50 user profile names can be specified.

Top

Authority (AUT)

Specifies the authority to be given to the users specified on the **Users (USER)** parameter.

Single values

***CHANGE**
The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

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

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**
The user cannot access the workstation object.

***AUTL**
The public authority of the authorization list specified on the AUTL parameter is used for the public authority for the object.

Note: You can specify AUT(*AUTL) only when USER(*PUBLIC) is also specified.

Other values (up to 10 repetitions)

***OBJALTER**
Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJMGT**
Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

***OBJEXIST**
Object existence authority provides the authority to control the object's existence and ownership. If a user has special save system authority (*SAVSYS), object existence authority is not needed to perform save restore operations on the object.

***OBJOPR**
Object operational authority provides authority to look at the description of an object and use the object as determined by the data authority that the user has to the object.

***OBJREF**
Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

Data authorities

***ADD** Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

***DLT** Delete authority provides the authority to remove entries from an object.

***EXECUTE**
Execute authority provides the authority needed to run a program or locate an object in a library.

***READ**
Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

***UPD** Update authority provides the authority to change the entries in an object.

Top

Authorization list (AUTL)

Specifies the authorization list whose members are to be given authority for the object specified for the **Workstation object type (WSOTYPE)** parameter. You must have authorization list management (*AUTLMGT) authority for the specified authorization list.

This is a required parameter unless either the **Users (USER)** parameter or the **Reference workstation object (REFWSO)** parameter is specified.

Top

Reference workstation object (REFWSO)

Specifies the workstation object referred to for authorizations. These authorizations are given to the object specified for the **Workstation object type (WSOTYPE)** parameter. Users authorized to the reference object are authorized in the same manner to the object for which authority is to be given. If the reference object is secured by an authorization list, that authorization list secures the object specified on the WSOTYPE parameter.

This is a required parameter unless either the **Users (USER)** parameter or the **Authorization list (AUTL)** parameter is specified.

Top

Examples

```
GRTWSOAUT   WSOTYPE(*TPLWRKARA)  AUTL(KLIST)
```

This command gives authority to the work are template to the users with authority specified for them on the authorization list KLIST.

Top

Error messages

Unknown

Top

Hold Communications Device (HLDCMNDEV)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Communications Device (HLDCMNDEV) command allows the operator to hold communication through the specified device. Communications are restarted with the Release Communications Device (RLSCMNDEV) command or by varying the device off and then on with the Vary Configuration (VRYCFG) command.

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.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	*Name*	Required, Positional 1
OPTION	Option	***CNTRLD**, *IMMED	Optional, Positional 2

Top

Device (DEV)

Specifies the name of the device whose communications are to be held. Devices whose communications are held are:

DEV Value	**Device**
3180	Display station
3277	Display station
3278	Display station
3279	Display station
3287	Printer (work station)
5219	Printer (work station)
5224	Printer (work station)
5225	Printer (work station)
5251	Display station
5252	Display station
5256	Printer (work station)
5291	Display station

5292	Display station
PLU1	Primary logical unit, type 1 (for SNA)
BSC	Binary synchronous device (Base and RJE)
BSCT	This &sys. system as a BSC multipoint tributary station
APPC	Logical unit in advanced program-to-program communications (APPC) network

This is a required parameter.

Top

Option (OPTION)

Specifies the manner in which communication is held with this device.

The possible values are:

***CNTRLD**
The specified device is not capable of communications at the next OPEN or ACQUIRE operation. The controlled option allows any program using the communications device to continue to do input/output operations, but no new uses of the device are started.

***IMMED**
The specified device is not capable of communications at the next READ, WRITE, OPEN, or ACQUIRE operation. The immediate option stops a communications device immediately, and a permanent input/output error is sent to the program.

Top

Examples

```
HLDCMNDEV   DEV(WSPR05)
```

This command holds the communications capability of the device WSPR05 at the time of the next OPEN or ACQUIRE operation.

Top

Error messages

***ESCAPE Messages**

CPF5920
Device &1 varied off or in diagnostic mode.

CPF5921
Device &1 not a communications device.

CPF5922
Device &1 already held with option *IMMED.

CPF5935
Error occurred during command processing.

CPF5984
Not authorized to perform function.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

Top

Hold Distribution Queue (HLDDSTQ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Distribution Queue (HLDDSTQ) command prevents a distribution queue from being sent.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	*Character value*	Required, Positional 1
PTY	Priority	*NORMAL, *HIGH	Required, Positional 2

Top

Distribution queue (DSTQ)

Specifies the name of the distribution queue that is held. The queue must have been previously configured using the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

This is a required parameter.

Top

Priority (PTY)

Specifies whether the normal priority or high priority portion of the specified queue is held.

The possible values are:

***NORMAL**
The normal priority queue is for those distributions with a service level of data low.

***HIGH**
The high priority queue is for those distributions with a service level of fast, status, or data high.

Note: This value is not valid for a SystemView distribution services (SVDS) type of distribution queue.

This is a required parameter.

Top

Examples

Example 1: Holding the Normal Priority Portion of a Queue

```
HLDDSTQ   DSTQ(CHICAGO)  PTY(*NORMAL)
```

This command holds the normal priority portion of the CHICAGO distribution queue.

Example 2: Holding the High Priority Portion of a Queue

```
HLDDSTQ   DSTQ(ATLANTA)  PTY(*HIGH)
```

This command holds the high priority portion of the ATLANTA distribution queue.

Top

Error messages

***ESCAPE Messages**

CPF8802
Distribution queue &1 was not found.

CPF8805
Special value for System name/Group not permitted or not used correctly.

CPF8806
Value &1 not valid for system name or system group.

CPF881C
High priority queue not allowed for *SVDS distribution queue &1

CPF8812
Error occurred while processing distribution queues.

CPF8816
QSNADS communications subsystem is not active.

CPF8817
Distribution queue is held.

CPF9845
Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

Top

Hold Job (HLDJOB)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Job (HLDJOB) command makes a job ineligible for processing by the system. The job is held until it is:

- Released by the Release Job (RLSJOB) command
- Cleared by the Clear Job Queue (CLRJOBQ) command
- Ended by the End Job (ENDJOB) command
- Ended (while the job is active) by the End Subsystem (ENDSBS) command, the End System (ENDSYS) command, or the Power Down System (PWRDWNSYS) command

Holding a job causes all threads within the job to be held.

Note: If you use this command to hold a job that has exclusive access to any resources on the system, these resources are not available to other jobs. Other jobs which require access to those resources will either fail or wait indefinitely.

Restrictions: The issuer of the command must be running under a user profile which is the same as the job user identity of the job being held, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority.

The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management book.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	*Qualified job name*	Required, Positional 1
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
SPLFILE	Hold spooled files	***NO**, *YES	Optional, Positional 2
DUPJOBOPT	Duplicate job option	***SELECT**, *MSG	Optional

Top

Job name (JOB)

Specifies the name of the job being held.

This is a required parameter.

Qualifier 1: Job name

name Specify the name of the job.

Qualifier 2: User

name Specify the user name that identifies the user profile under which the job is started.

Qualifier 3: Number

000000-999999
Specify the system-assigned job number.

Note: If no user name or job number is specified, all jobs currently in the system are searched for the job name. If more than one occurrence of the specified name is found, a qualified job name must be provided either explicitly or through the selection display. Refer to the **Duplicate job option (DUPJOBOPT)** parameter for more information.

Top

Hold spooled files (SPLFILE)

Specifies whether spooled output files created by the job being held are also held.

***NO** The spooled output files produced by the job are not held.

***YES** The spooled output files produced by the job are also held. Only those spooled output files which are on output queues in the library name space of the thread issuing this command will be held. If the **Spooled file action (SPLFACN)** job attribute is *DETACH and the job is ended while the spooled files are held, the spooled files cannot be released using the Release Job (RLSJOB) command. To release spooled files after the job has been removed from the system, use the Release Spooled File (RLSSPLF) command.

Top

Duplicate job option (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. Otherwise, a message is issued.

***MSG** A message is issued when duplicate jobs are found.

Top

Examples

Example 1: Making a Job Ineligible for Processing

```
HLDJOB   JOB(PAYROLL)  SPLFILE(*YES)
```

This command makes the job named PAYROLL ineligible for processing. All spooled files for this job are also held.

Example 2: Holding a Job that has a Duplicate Name

```
HLDJOB   JOB(DEPTXYZ/PAYROLL)
```

This command holds the job named PAYROLL submitted by a user operating under the user profile DEPTXYZ. The qualified form of the job name is used when jobs with duplicate names exist in the system. Spooled files are not held.

Top

Error messages

***ESCAPE Messages**

CPF1E52
Not authorized to hold job &1.

CPF1E53
Job &1 has ended and cannot be held.

CPF1E54
Job &1 cannot be held.

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1321
Job &1 user &2 job number &3 not found.

CPF1332
End of duplicate job names.

CPF1340
Job control function not performed.

CPF1341
Reader or writer &3/&2/&1 not allowed as job name.

CPF1342
Current job not allowed as job name on this command.

CPF1343
Job &3/&2/&1 not valid job type for function.

CPF1344
Not authorized to control job &3/&2/&1.

CPF1345
Cannot hold job &3/&2/&1.

CPF1346
Job &3/&2/&1 already held.

CPF1347
Cannot hold job &3/&2/&1.

CPF1348
Job &3/&2/&1 held but unable to hold its files.

CPF1350
SPLFILE(*NO) specified but job &3/&2/&1 on OUTQ.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1352

Function not done. &3/&2/&1 in transition condition.

CPF1378

Job &3/&2/&1 not held at current time.

Top

Hold Job Queue (HLDJOBQ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Job Queue (HLDJOBQ) command prevents the processing of all jobs currently waiting on the job queue and of all jobs that are added to the queue after the command is issued. This command has no effect on jobs that are running. Additional jobs can be placed on the job queue while it is held, but they are not processed. The jobs are held until a Release Job Queue (RLSJOBQ) command is issued. When a job queue is held, the jobs can be cleared with the Clear Job Queue (CLRJOBQ) command or a specific job can be canceled by the End Job (ENDJOB) command.

Restriction: The QLPINSTALL job queue cannot be held.

Top

Parameters

Keyword	Description	Choices	Notes
JOBQ	Job queue	*Qualified object name*	Required, Positional 1
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Job queue (JOBQ)

Specifies the job queue that will have its current and future entries withheld from further processing.

This is a required parameter.

Qualifier 1: Job queue

name Specify the name of the job queue to be held.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

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

Top

Examples

```
HLDJOBQ   JOBQ(QBATCH)
```

This command prevents the processing of the jobs currently on the QBATCH job queue and any jobs added to the queue. They are held until the queue is released or cleared. Individual jobs can also be ended with the ENDJOB command, which removes the job from the job queue.

Top

Error messages

***ESCAPE Messages**

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF2240
User &7 not authorized to use *&5 &6/&4.

CPF3307
Job queue &1 in &2 not found.

CPF3330
Necessary resource not available.

CPF3425
Job queue &1 in &2 already held.

Top

Hold Job Schedule Entry (HLDJOBSCDE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Job Schedule Entry (HLDJOBSCDE) command allows you to hold an entry, all entries, or a set of entries in the job schedule. Each job schedule entry contains the information needed to automatically submit a job to be run once or at regularly scheduled intervals.

If you hold a job schedule entry:

- The entry is held until it is released using the Release Job Schedule Entry (RLSJOBSCDE) or Work with Job Schedule Entries (WRKJOBSCDE) command.
- The job is not submitted when it is released, even if a date and time at which it was scheduled to be submitted passed while the entry was held.

Restrictions:

1. To hold entries, you must have job control (*JOBCTL) special authority; otherwise you can hold only those entries that you added.
2. To use this command, you must have:
   - Use (*USE) authority to object QDFTJOBSCD, type *JOBSCD, in library QUSRSYS and execute (*EXECUTE) authority to library QUSRSYS.

Top

Parameters

Keyword	**Description**	**Choices**	**Notes**
JOB	Job name	*Generic name, name*, *ALL	Required, Positional 1
ENTRYNBR	Entry number	000001-999999, ***ONLY**, *ALL	Optional

Top

Job name (JOB)

Specifies the name of the job schedule entry.

This is a required parameter.

***ALL** All of the job schedule entries for which you have authority are held. If JOB(*ALL) is specified, ENTRYNBR(*ALL) must also be specified.

generic-name
Specify the generic name used to find job schedule entries. 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 entries with names that begin with the generic name, and for which the user has authority, are held. If a generic name is specified, ENTRYNBR(*ALL) must also be specified. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete job name.

name Specify the name of the job schedule entry that you want held.

Top

Entry number (ENTRYNBR)

Specifies the number of the job schedule entry you want held. The message sent when an entry is successfully added contains the entry number. You can also determine the entry number by using the Work with Job Schedule Entries (WRKJOBSCDE) command. Press F11 from the WRKJOBSCDE display to show the entry numbers of the selected entries.

***ONLY**
Only one entry in the job schedule has the job name specified for the JOB parameter. If *ONLY is specified and more than one entry has the specified job name, no entries are held and an error message is sent.

***ALL** All entries with the specified job name are held.

000001-999999
Specify the number of the job schedule entry you want held.

Top

Examples

Example 1: Holding a Job Schedule Entry

```
HLDJOBSCDE   JOB(CLEANUP)
```

This command holds the job schedule entry with the job name CLEANUP.

Example 2: Holding All Job Schedule Entries

```
HLDJOBSCDE   JOB(*ALL)  ENTRYNBR(*ALL)
```

This command holds all entries in the job schedule.

Example 3: Holding an Individual Job Schedule Entry

```
HLDJOBSCDE   JOB(PAYROLL)  ENTRYNBR(*ONLY)
```

This command holds the entry PAYROLL in the job schedule.

Example 4: Holding a Generic Job Schedule Entry

```
HLDJOBSCDE   JOB(PAY*)  ENTRYNBR(*ALL)
```

This command holds all entries in the job schedule that have the prefix PAY in their names.

Top

Error messages

***ESCAPE Messages**

CPF1628
Job schedule entry &3 number &4 not found.

CPF1629
Not authorized to job schedule &1.

CPF1630
Not authorized to job schedule entry &3 number &4.

CPF1632
Job schedule entry &3 number &4 damaged.

CPF1636
More than one entry with specified entry job name found.

CPF1637
Job schedule &1 in library &2 in use.

CPF1638
Job schedule entry &3 number &4 in use.

CPF1640
Job schedule &1 in library &2 does not exist.

CPF1641
Job schedule &1 in library &2 damaged.

CPF1645
No job schedule entries found for specified name.

CPF1646
Entry number must be *ALL when generic name specified.

CPF1647
&3 entries successfully held, &4 entries not held.

CPF1649
Entry number must be *ALL.

Top

Hold Output Queue (HLDOUTQ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Output Queue (HLDOUTQ) command prevents all currently waiting spooled files, and all spooled files that are added to the output queue after the command is issued, from being processed by a spooling writer. This command has no effect on jobs currently running and adding spooled files to the output queue. It also has no effect on the spooled output that is being produced by a spooling writer at the time the command is issued. When the spooling writer completes all copies of the current output file, it cannot begin the output for any other files until the queue is released.

Top

Parameters

Keyword	Description	Choices	Notes
OUTQ	Output queue	*Qualified object name*	Required, Positional 1
	Qualifier 1: Output queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Output queue (OUTQ)

Specifies the name of the output queue that will have its current and future spooled files withheld from further processing.

This is a required parameter.

Qualifier 1: Output queue

name Specify the name of the output queue to hold.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.

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

Top

Examples

```
HLDOUTQ   OUTQ(QPRINT)
```

This command prevents the processing of the spooled files on the QPRINT queue and any spooled files added to the queue. They are held until the queue is released or cleared. A specific job (with its spooled files) can also be ended with the ENDJOB command, which removes the spooled files from the output queue.

Top

Error messages

***ESCAPE Messages**

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF3330
Necessary resource not available.

CPF3357
Output queue &1 in library &2 not found.

CPF3426
Output queue &1 in library &2 already held.

Top

Hold Reader (HLDRDR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Reader (HLDRDR) command immediately stops the activity of the specified spooling reader. The reader itself is not ended, nor is its associated input device made available to the system. The reader remains inactive until a Release Reader (RLSRDR) or End Reader (ENDRDR) command is issued. Data is not lost when the reader is held.

Top

Parameters

Keyword	Description	Choices	Notes
RDR	Reader	*Name*	Required, Positional 1

Top

Reader (RDR)

Specifies the spooling reader to be held.

This is a required parameter.

name Specify the name of the reader to be held.

Top

Examples

```
HLDRDR   RDR(QDKT)
```

This command causes the diskette reader QDKT to immediately stop reading data. To release the reader, so that it can continue to read data, a Release Reader (RLSRDR) command must be entered. If the End Reader (ENDRDR) command is used, the reader is stopped and the job that was being read in is lost because no job entry was added to the job queue.

Top

Error messages

*ESCAPE Messages

CPF1E52
Not authorized to hold job &1.

CPF1E53
Job &1 has ended and cannot be held.

CPF1E54
Job &1 cannot be held.

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1340
Job control function not performed.

CPF1345
Cannot hold job &3/&2/&1.

CPF1347
Cannot hold job &3/&2/&1.

CPF1350
SPLFILE(*NO) specified but job &3/&2/&1 on OUTQ.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1352
Function not done. &3/&2/&1 in transition condition.

CPF1378
Job &3/&2/&1 not held at current time.

CPF3312
Reader &1 neither active nor on job queue.

CPF3330
Necessary resource not available.

CPF3333
Reader &3/&2/&1 already held.

CPF3490
Not authorized to specified reader.

Top

Hold Spooled File (HLDSPLF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Spooled File (HLDSPLF) command stops the specified spooled file from additional processing by a spooled writer. If the file is being produced on an output device, the writer stops processing that file and gets the next file to be processed. When the file is released and selected for output, it is again processed starting at the beginning of the file. If multiple copies are being produced for the file when it is held, the incomplete copy is produced from the beginning again and any remaining copies follow it.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Spooled file	*Name*, *SELECT	Required, Positional 1
JOB	Job name	Single values: * Other values: *Qualified job name*	Optional, Positional 2
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, ***ONLY**, *LAST, *ANY	Optional, Positional 3
JOBSYSNAME	Job system name	*Name*, ***ONLY**, *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: ***ONLY**, *LAST Other values: *Element list*	Optional
	Element 1: Creation date	*Date*	
	Element 2: Creation time	*Time*, ***ONLY**, *LAST	
SELECT	Select files for	*Element list*	Optional
	Element 1: User	*Name*, ***CURRENT**, *ALL	
	Element 2: Print device	*Name*, ***ALL**, *OUTQ	
	Element 3: Form type	*Character value*, ***ALL**, *STD	
	Element 4: User data	*Character value*, ***ALL**	
	Element 5: ASP	1-32, ***ALL**, *ASPDEV	
ASPDEV	ASP device	*Name*, *, *SYSBAS, *CURASPGRP	Optional
OPTION	When to hold file	***IMMED**, *PAGEEND	Optional

Top

Spooled file (FILE)

Specifies the spooled file to hold.

This is a required parameter.

***SELECT**
All spooled files that meet the selection values specified on the **Select files for (SELECT)** parameter are held. This value is mutually exclusive with values specified on the **Job name (JOB)** parameter, **Spooled file number (SPLNBR)** parameter, **Job system name (JOBSYSNAME)** parameter, and **Spooled file created (CRTDATE)** parameter.

name Specify the name of the spooled file to hold.

Top

Job name (JOB)

Specifies the job that created the file being held.

Single values

***** The job that issued this Hold Spooled File (HLDSPLF) command is the job that produced this file.

Qualifier 1: Job name

name Specify the name of the job that created the file being held.

Qualifier 2: User

name Specify the user name that identifies the user profile under which the job is run.

Qualifier 3: Number

000000-999999
Specify the system-assigned job number.

Top

Spooled file number (SPLNBR)

Specifies the number of the spooled file that was created by the specified job.

***ONLY**
Only one spooled file in the job has the specified file name; therefore, the number of the spooled file is not necessary.

***LAST**
The spooled file with the highest number and the specified file name is used.

***ANY** The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file create date and time parameter is to take precedence over the spooled file number when selecting a spooled file.

1-999999
Specify the number of the spooled file having the specified file name that is being held.

Top

Job system name (JOBSYSNAME)

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

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

***ANY** The job system name is not used to determine which spooled file is used. Use this value when the spooled file create date and time parameter is to take precedence over the job system name when selecting a spooled file.

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

Top

Spooled file created (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.

Single values

***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 create 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: Creation date

date Specify the date the spooled file was created.

Element 2: Creation time

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

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

time Specify the time the spooled file was created.

Top

Select files for (SELECT)

Specifies which group of files are to be held. Positional values can be specified to select the files: the user that created the file, the device that the file is queued for, the form type specified, the user data tag associated with the file, or the auxiliary storage pool the file is in. Only files that meet each of the values are selected.

Element 1: User

***CURRENT**
Only files created by the user running this command are selected.

***ALL** Files created by all users are selected.

name Specify a user name. Only files created by that user name are selected.

Element 2: Print device

***ALL** Files on any device-created or user-created output queue are selected.

***OUTQ**
All files on any user-created output queue are selected. A user-created output queue is any output queue that is not automatically created by a device. A user-created output queue does not generally have the same name as a device, but if it does, it does not reside in library QUSRSYS.

name Specify a device name. Only files on the device created output queue for that device are selected. A device created output queue is one that has the same name as a device and resides in the QUSRSYS library. Unless it already exists, it will automatically be created by the system when the device is created. A device created output queue cannot be deleted.

Element 3: Form type

***ALL** Files for all form types are selected.

***STD** Only files that specify the standard form type are selected.

form-type
Specify the form type to select the file.

Element 4: User data

***ALL** Files with any user data tag specified are selected.

user-data
Specify the user data tag to select the file.

Element 5: ASP

***ALL** All files as specified in the Auxiliary Storage Pool Device (ASPDEV) parameter are selected.

***ASPDEV**
Files specified in the Auxiliary Storage Pool Device (ASPDEV) parameter are selected.

1-32 Specify the auxiliary storage pool (ASP) of the files being selected.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool device name from which spooled files are to be selected. This parameter is only valid if the ASP number (ASP) element of the Select parameter is *ALL or *ASPDEV.

***** Files which are found in the ASPs that are currently part of the thread’s library name space are selected. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and if the thread has an ASP group, the primary and secondary ASPs in the thread’s ASP group.

***SYSBAS**
Files which are found in the system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) are selected.

***CURASPGRP**
Files which are found in the primary and secondary ASPs in the thread's ASP group are selected. If no ASP group is associated with the thread, an error will be issued.

name Specify the name of the auxiliary storage pool device description. Files which are found in the specified primary or secondary ASP are selected. Only primary or secondary ASPs which are in the thread's ASP group may be specified. If no ASP group is associated with the thread, an error will be issued.

Top

When to hold file (OPTION)

Specifies which option to use when holding the spooled file.

Note: Specifying an option when the file is not being written has no effect.

***IMMED**
The file is to be held as soon as possible.

***PAGEEND**
The file is to be held on a page boundary.

Top

Examples

Example 1: Holding a File Created by Another Job

```
HLDSPLF   FILE(SHIPITEMS)  JOB(00009/JONES/ORDER)
```

This command withholds the spooled file SHIPITEMS, created by the job ORDER, from additional processing.

Example 2: Holding a File at a Page Boundary

```
HLDSPLF   FILE(QPJOBLOG)  OPTION(*PAGEEND)
```

This command holds the spooled file QPJOBLOG at a page boundary.

Example 3: Holding a File Immediately

```
HLDSPLF   FILE(QPJOBLOG)  OPTION(*IMMED)
```

This command holds the spooled file QPJOBLOG immediately. Holding a spooled file by specifying this option causes the CHGSPLFA command RESTART(*NEXT) to be inaccurate if the spooled file is currently being processed by a spool writer.

Top

Error messages

***ESCAPE Messages**

CPF337E
ASP device &1 not in current ASP group for thread.

CPF337F
ASP device &1 not allowed with ASP number &2.

CPF33D0
Printer &1 does not exist.

CPF33D1
User &1 does not exist.

CPF3303
File &1 not found in job &5/&4/&3.

CPF3309
No files named &1 are active.

CPF3330
Necessary resource not available.

CPF3337
File &1 number &8 already held or saved.

CPF3340
More than one file with specified name found in job &5/&4/&3.

CPF3342
Job &5/&4/&3 not found.

CPF3343
Duplicate job names found.

CPF3344
File &1 number &8 no longer in the system.

CPF3357
Output queue &1 in library &2 not found.

CPF34A4
File &1 number &8 not held or deleted.

CPF3492
Not authorized to spooled file.

CPF9825
Not authorized to device &1.

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

CPFB8ED
Device description &1 not correct for operation.

Top

Hold Writer (HLDWTR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Hold Writer (HLDWTR) command stops the specified writer at the end of a record, at the end of a spooled file, or at the end of a printed page. If multiple copies of a file are produced, the writer can be held at the end of the copy currently being produced. The writer is not stopped and the device is not made available to the system. The writer remains inactive until a Release Writer (RLSWTR) or End Writer (ENDWTR) command is issued. Data is not lost when the writer is held.

Top

Parameters

Keyword	Description	Choices	Notes
WTR	Writer	*Name*	Required, Positional 1
OPTION	When to hold writer	***IMMED**, *CNTRLD, *PAGEEND	Optional, Positional 2

Top

Writer (WTR)

Specifies the spooling writer being held.

This is a required parameter.

name Specify the name of the writer to be held.

Top

When to hold writer (OPTION)

Specifies when the spooling writer should stop producing output.

***IMMED**

The writer stops immediately after it has written the last record, in the current block of records, to the output device. Each time the writer finishes producing a block of records on a device, it makes another I/O request to get the next block from the file being spooled to the device. If *IMMED is specified, the writer stops only after it has written the last record in the block being processed, which (for diskette output) is a complete diskette record being written on diskette.

When *IMMED is specified for printed output, the writer stops anywhere within or at the end of a print line or at the end of a complete block, which may not be at the end of a line. This is because some data records (which are blocked to improve performance) may be split in two, with the first part of a record at the end of one block and the last part of the record at the beginning of the next block. If only one copy of the file is being produced or if the last copy is being produced, the entry for the file is removed from the output queue when the output is completed.

***CNTRLD**

The writer stops at the end of the current copy of the file. If only one copy of the file is to be produced or if the last copy is being produced, the entry for the file is removed from the output queue when the output is completed.

***PAGEEND**

The writer is held at the end of a page. This value is valid only when the spooling writer is a printer writer.

Top

Examples

```
HLDWTR   WTR(PRINTER)  OPTION(*CNTRLD)
```

This command stops the writer named PRINTER at the end of the current file. The writer is held until an RLSWTR (Release Writer) or ENDWTR (End Writer) command is issued.

Top

Error messages

***ESCAPE Messages**

CPF1340

Job control function not performed.

CPF3313

Writer &1 not active nor on job queue.

CPF3330

Necessary resource not available.

CPF3331

Not authorized to control writer &3/&2/&1.

CPF3332

Writer &3/&2/&1 already held.

CPF3334

Previous hold to writer &3/&2/&1 pending.

CPF3438

*PAGEEND not valid for writer &3/&2/&1.

Top

Start HOST Query (HOST)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Start HOST Query (STRHOSTQRY) command, or its alias HOST, is a simple utility for performing Domain Name System (DNS) lookups. It is normally used to convert names to IP addresses and vice versa.

Restrictions:

- You must have execute (*X) authority to the directories in the path of the output file.
- You must have write (*W) authority to the output file if it already exists.
- You must have read, write and execute (*RWX) authority to the output file's parent directory if the output file does not already exist.

Top

Parameters

Keyword	Description	Choices	Notes
HOSTNAME	Query name	*Character value*	Required, Positional 1
TYPE	Query type	***A**, *AAAA, *ANY, *AXFR, *CNAME, *MX, *NS, *PTR, *SOA, *SRV, *TXT	Optional, Positional 2
CLASS	Query class	***IN**, *CH, *HS, *ANY	Optional
DMNNAMSVR	Domain name server	*Character value*, ***CFG**	Optional
SOA	Display SOA records	***NO**, *YES	Optional
AXFR	List all hosts	***NO**, *YES	Optional
IP6INT	Use IP6.INT domain	***NO**, *YES	Optional
SETRDFLAG	Recursion desired	***YES**, *NO	Optional
PROTOCOL	Network protocol	***UDP**, *TCP	Optional
DEBUG	Show debug information	***NO**, *YES	Optional
IPVSN	IP Version	***ALL**, *IPV4ONLY, *IPV6ONLY	Optional
NBRDOTS	Number of dots	0-10, **1**	Optional
TIMEOUT	Query timeout	1-100, **5**	Optional
UDPNBRRTY	UDP retry	0-100, **2**	Optional
TOSTMF	Output file	*Path name*, ***STDOUT**	Optional

Top

Query name (HOSTNAME)

Specifies the name that you want the Domain Name System (DNS) server to look up. It can also be a dotted-decimal IPv4 address or a colon-delimited IPv6 address, in which case HOST will by default perform a reverse lookup for that address.

This is a required parameter.

domain-name
Specify a valid domain name.

internet-address
Specify a valid IPv4 or IPv6 address.

Top

Query type (TYPE)

Specifies the type of the query. The default type is *A. The list of supported query types changes with time, and not all servers support all the types that exist. This list provides some common query types for convenience and is not a complete list. If you do not see a query type you want in this list, you can still type in a character string that represents that query type. If the type is unknown by this BIND version, the query will default to an *A query with any corresponding results.

<u>*A</u> IPv4 Address record. This is the character string ’a’.

***AAAA**
IPv6 address record. This is the character string ’aaaa’.

***ANY** Any resource record. This is the character string ’any’.

***AXFR**
Zone transfer. This is the character string ’axfr’.

***CNAME**
Canonical name record. Returns a list of aliases for the true (canonical) host name, if any exist. This is the character string ’cname’.

***MX** Mail exchange record. This is the character string ’mx’.

***NS** Name server (DNS server) information for the zone. This is the character string ’ns’.

***PTR** Pointer record. Returns a name for an IP address. This is the character string ’ptr’.

***SOA** Start of authority record. This is the character string ’soa’.

***SRV** Services location selection. This is the character string ’srv’.

***TXT** Text record. This is the character string ’txt’.

Top

Query class (CLASS)

Specifies the protocol group of the information.

<u>*IN</u> The Internet class.

***CH** The CHAOS class. The server provides some helpful diagnostic information through a number of built-in zones under the CHAOS class.

version
If your BIND server named.conf has the ’version’ option configured, then you can query for it using the CHAOS class. This is the version the server should report via a query of the name ’version.bind’ with type TXT, class CHAOS. The default is the real version number of this server.

```
HOST   HOSTNAME('version.bind')  TYPE(*TXT)  CLASS(*CH)
```

hostname
If your BIND server named.conf has the ’hostname’ option configured, then you can query for it using the CHAOS class. This is the hostname the server should report via a

query of the name ’hostname.bind’ with type TXT, class CHAOS. This defaults to the hostname of the machine hosting the name server. The primary purpose of such queries is to identify which of a group of anycast servers is actually answering your queries.

```
STRHOSTQRY   NAME('hostname.bind')  TYPE(*TXT)  CLASS(*CH)
```

server-id
If your BIND server named.conf has the ’server-id’ option configured, then you can query for it using the CHAOS class. This is the ID of the server should report via a query of the name ’ID.SERVER’ with type TXT, class CHAOS. The primary purpose of such queries is to identify which of a group of anycast servers is actually answering your queries. The default server-id is none.

```
HOST   HOSTNAME('ID.SERVER')  TYPE(*TXT)  CLASS(*CH)
```

***HS** The MIT Athena Hesiod class. Hesiod, developed by MIT Project Athena, is an information service built upon BIND. Its intent is to furnish information about users, groups, network-accessible file systems, printcaps, and mail service throughout an installation. In other words, it holds arbitrary data stored as text strings. Queries that use the HS class retrieve data stored as TXT Resource Records. You cannot specify root servers for the Hesiod class (generally the ones at MIT) in your list of root name servers.

***ANY** Query any class. This is a wildcard query.

Top

Domain name server (DMNNAMSVR)

Specifies the name or the IP address of the DNS server that HOST will use as its current server for the query session. You can specify any DNS server to which your TCP/IP network has access.

HOST retrieves information from DNS servers. It needs an active DNS server to send its queries to. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to set one of the following as its default DNS server for the session:

1. DNS server your system is configured to use (*CFG), or
2. The DNS server that is running on your local system.

***CFG** Use the DNS server that is currently designated for use by this system. These server internet addresses can be seen by prompting the Change TCP/IP Domain (CHGTCPDMN) command and looking at the values shown for the INTNETADR parameter.

server-domain-name
Specify the name of a DNS server.

server-internet-address
Specify the IP address of a DNS server.

Top

Display SOA records (SOA)

Specifies whether or not to print the SOA records for zone name from all the listed authoritative name servers for that zone. The list of name servers is defined by the NS records that are found for the zone.

***NO** Do not print SOA records.

***YES** Attempt to print SOA records.

Top

List all hosts (AXFR)

Specifies whether or not to make HOST perform a zone transfer for zone specified in the HOSTNAME parameter. A zone transfer is when all domain information in the zone is returned. HOST will display the NS, PTR and address records (A/AAAA).

***NO** Do not perform a zone tranfer.

***YES** Perform a zone transfer.

Top

Use IP6.INT domain (IP6INT)

Specifies whether or not to qualify the reverse lookup to be in the ip6.int zone and not the ip6.arpa zone.

***NO** The normal ip6.arpa zone reverse lookup will be performed.

***YES** An IPv6 address reverse lookup in the ip6.int zone will be performed. This zone is deprecated, but may still be required to query IPv6 backbone prefixes.

Top

Recursion desired (SETRDFLAG)

Specifies whether or not to set the Recursion Desired (RD) flag in the query. This should mean that the name server receiving the query will not attempt to resolve name. This enables HOST to mimic the behaviour of a name server by making non-recursive queries and expecting to receive answers to those queries that are usually referrals to other name servers.

***YES** Set the RD flag.

***NO** Do not set the RD flag.

Top

Network protocol (PROTOCOL)

Specified whether to use TCP or UDP when sending requests to the server.

***UDP** Use UDP to send the query. However, TCP will be automatically selected for queries that require it, such as zone transfer (AXFR) requests.

***TCP** Use TCP to send the query.

Top

Show debug information (DEBUG)

Specifies whether or not to turn debugging mode on. More information is displayed about the packet sent to the server and the resulting answer when debugging mode is on.

***NO** Turn off debugging messages.

***YES** Turn on debugging messages.

Top

IP Version (IPVSN)

Specifies whether to limit the query to IPv4 or IPv6 networks.

***ALL** Do not limit queries to IPv4 or IPv6.

***IPV4ONLY**
Only send queries out IPv4 network interfaces.

***IPV6ONLY**
Only send queries out IPv6 network interfaces.

Top

Number of dots (NBRDOTS)

Specifies the number of dots (period characters) that have to appear in a domain name for it to be considered absolute. Names with fewer dots are interpreted as relative names and will be searched for in the domains listed in the DMNSCHLIST parameter.

1 If the domain name contains one dot it is considered an absolute name.

0-10 Specify the number of dots that must be in a domain name for the domain name to be considered to be an absolute name.

Top

Query timeout (TIMEOUT)

Specifies the timeout interval, in seconds, to wait for a response.

5 A timeout value of 5 seconds is used.

1-100 Specify a valid timeout value in seconds.

Top

UDP retry (UDPNBRRTY)

Specifies the number of times to retry UDP queries to the current DNS server before attempting TCP queries.

2 Two UDP retries will be used. This is in addition to the initial query.

0-100 Specify the number of UDP retries. You normally only need to retry a query 3 to 5 times.

Top

Output file (TOSTMF)

Specifies the name of a stream file where all command output is written.

***STDOUT**
All command output goes to the standard output device (normally the display).

path-name
Specify the path for a stream file where output should be written.

Top

Examples

Example 1: Looking Up Internet Address for Domain Name

```
STRHOSTQRY  HOSTNAME(ibm.com)
```

This command attempts to find information about the domain 'aol.com'. This includes A and MX records, althought the output is formatted in a different style than other query tools (e.g. DIG). Sample output from this query might look like this:

```
Using domain server:
Name: 9.5.176.200
Address: 9.5.176.200#53
Aliases:
aol.com has address 64.12.50.151
aol.com has address 205.188.142.182
aol.com mail is handled by 15 mailin-01.mx.aol.com.
aol.com mail is handled by 15 mailin-02.mx.aol.com.
aol.com mail is handled by 15 mailin-03.mx.aol.com.
aol.com mail is handled by 15 mailin-04.mx.aol.com.
```

Example 2: Lookup with a Zone Transfer

```
HOST   HOSTNAME(i5os.ibm.com)  AXFR(*YES)
```

This command attempts to do a zone transfer. Sample output from this query might look like this:

```
Using domain server:
Name: 10.0.1.100
Address: 10.0.1.100#53
Aliases:
i5os.ibm.com name server MYDNS1.IBM.COM.
i5os.ibm.com name server MYDNS2.IBM.COM.
i5os.ibm.com has address 10.0.1.100
i5os.ibm.com has address 10.0.2.200
box1.i5os.ibm.com has address 10.0.2.201
box2.i5os.ibm.com has address 10.0.2.202
```

This command attempts to do the same zone transfer, but fails because the user is not authorized to do transfers. Sample output from this query might look like this:

```
Using domain server:
Name: 10.0.1.100
Address: 10.0.1.100#53
Aliases:
Host i5os.ibm.com not found: 9(NOTAUTH)
; Transfer failed.
```

Top

Error messages

*ESCAPE Messages

DNS0013
Error processing command parameters.

DNS0065
Option 33 of i5/OS is required, but is not installed.

TCP7124
Program &1 in library &2 type *PGM ended abnormally.

Top

If (IF)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The If (IF) command evaluates a logical expression and conditionally processes CL program or ILE CL procedure commands according to the evaluation of the expression. If the logical expression is true (a logical 1), the command (or the group of commands in a Do group) specified in the THEN parameter is processed, and the ELSE command with its associated command or Do group is not processed. If the result of the logical expression is false (a logical 0), the command specified in the THEN parameter is not processed and control passes to the next command. If that command is an ELSE command, the command or Do group specified in that command is processed. If the ELSE command is not specified, control passes to the next command.

When a DO command is specified, either in the THEN parameter of the IF command or in the CMD parameter of the ELSE command, the Do group is bypassed if the result of the expression is not the one needed for the group being processed. That is, control passes to the command that follows the ENDDO command associated with that DO.

When the command or Do group specified by the THEN parameter or the ELSE command is completed (and no GOTO command has been processed), control passes to the next command following the command or Do group specified by the ELSE command. If a GOTO command is processed, control is passed to the command with the label specified by the GOTO command, and processing continues from that command.

The following command sequence shows this flow. In this example, &TESTSW is a logical variable.

```
IF &TESTSW DO
   Group A  (group of CL commands)
        .
        .
   ENDDO
ELSE DO
   Group B  (group of CL commands)
        .
        .
   ENDDO
Group C  (continued CL commands)
        .
        .
```

The IF command tests the logical variable &TESTSW. If a true condition exists (&TESTSW contains a value of '1'), the commands in Group A are processed, then control passes to the commands in Group C. If a false condition exists (&TESTW contains a value of 0), the commands in group A are bypassed, the commands in Group B are processed, then control passes to the commands in Group C.

Restrictions:

- The IF command is valid only in a CL program or ILE CL procedure.
- Up to ten levels of nested IF and ELSE commands are allowed.

Top

Parameters

Keyword	Description	Choices	Notes
COND	Condition	*Logical value*	Required, Positional 1
THEN	Command	*Command string*	Optional, Positional 2

Top

Condition (COND)

Specifies the logical expression that is evaluated to determine a condition in the program and what is done next. Refer to ″Logical Expressions″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a description of logical expressions. Note that variables, constants, and the %SUBSTRING, %SWITCH, and %BINARY built-in functions can be used within the expression.

This is a required parameter.

logical-value
Specify the name of a CL logical variable or a logical expression.

Top

Command (THEN)

Specifies the command or group of commands (in a Do group) that are processed if the result of evaluating the logical expression is true. After the command or Do group is processed, control is passed to the next command *after* the ELSE command associated with this IF command. If the result is true, the ELSE command associated with the IF command is not processed. If the command specified in this parameter is a DO command, all commands within the Do group are considered to be the command specified by the parameter.

If the command specified by the THEN keyword is not coded on the same line when the keyword is coded, the THEN keyword must be immediately followed (on the same line) either by the left parenthesis or by a plus sign (+) or a minus sign (-) to show continuation. (A blank cannot immediately follow any keyword.) The command and the right parenthesis can then be coded on the next line. For example:

```
IF COND(&A *EQ &B)  THEN(      +
  GOTO C)
```

If any part of the command specified by the THEN parameter continues on the next line, a continuation character (+ or -) must be specified.

If a DO command is specified, only the DO command (not the commands specified within the Do group) is within the parentheses. For example:

```
IF COND(&A *EQ &B)  THEN(DO)
  CMD1
  CMD2
    .
    .
  ENDDO
```

If no command is specified for the THEN parameter (a null THEN) and the ELSE command immediately follows it, the ELSE is processed if the IF expression is false and it is skipped if the expression is true.

Any CL command can be specified for the THEN parameter, except the following commands:

- ELSE
- PGM, ENDPGM
- ENDDO
- MONMSG
- DCL, DCLF
- WHEN, OTHERWISE, ENDSELECT

The command can be another IF, unless there are already ten levels of nested IF and ELSE commands.

Top

Examples

```
IF   COND(&A *EQ &B)  THEN(GOTO X)
IF   (&A *EQ &B)  THEN(GOTO X)
IF   (&A *EQ &B)  (GOTO X)
IF   COND(&A *EQ &B)  THEN(GOTO X)
```

The examples above show a number of different ways the IF command can be specified to test a condition and branch to a label. In each of these examples, if &A equals &B, control passes to a CL command that has a label named X.

```
IF  COND(&TESTSW)  THEN(CHGVAR  VAR(&A)  VALUE(23))
```

If &TESTSW has a logical value of 1 (true), the CHGVAR command is processed to set &A to decimal 23; if &TESTW has a logical value of 0 (not true), the Change Variable (CHGVAR) command is not processed.

```
IF   COND((&ALPHA *EQ &BETA) *AND *NOT &GAMMA)
     THEN(RETURN)
```

If the value of &ALPHA equals the value of &BETA and if &GAMMA is a logical 0, then return to the program or procedure that called this CL program or ILE CL procedure.

```
IF   &LOG1  THEN(IF  (&A *GT 10) THEN(GOTO X))
                     ELSE(GOTO Y)
ELSE  DO
 :  (group of CL commands)
ENDDO
```

This is an example of nested IF commands. If &LOG1 has a logical value of 1 (true) and if &A is greater than 10, a branch is made to label X. If &LOG1 has a logical value of 1 and &A is *not* greater than 10, a branch is made to label Y. If &LOG1 has a logical value of 0 (false), &A is not compared to 10. Instead, the DO group of the second ELSE command is processed.

```
IF   &TEST  THEN(DO)
  CHGVAR  &A (&A + 1)
  GOTO   X
ENDDO
ELSE   DO
  CHGVAR   &B  (&B + 1)
  CALL   EXTPGM  (&B)
ENDDO
```

This example shows how the THEN parameter can be continued on the next line. If &TEST has a logical value of 1, the Do group specified in the THEN parameter is processed. Otherwise, the Do group specified by the ELSE command is processed.

```
IF   (&A *EQ YES)  DO
  CHGVAR   &B 1
  CHGVAR   &C 'Z'
ENDDO
```

This example shows a Do group as the THEN parameter. The two Change Variable (CHGVAR) commands are processed if, in the relational expression, &A is equal to YES.

```
IF   %SWITCH(10XXXX10)   THEN(GOTO X)
```

This example shows how the built-in function %SWITCH is used to test the eight job switches in a job. Refer to the topic ″Built-in functions for CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a complete description of %SWITCH. In this example, job switches 1, 2, 7, and 8 are tested for the values indicated in the 8-character mask. If switches 1 and 7 contain 1s and switches 2 and 8 contain 0s, then control branches to the command having the label X. If any of the four switches do not contain the value indicated, the branch does not occur.

Top

Error messages

*ESCAPE Messages

CPF0816

%SWITCH mask &1 not valid.

Top

Include CL Source (INCLUDE)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

Parameters
Examples
Error messages

The Include CL Source (INCLUDE) command includes external text into the source program being compiled. The external text must be valid CL commands that are valid in a compiled CL source. These commands can be declare statements (like DCL or DCLF), control flow statements (like IF or Select), or regular CL commands (like CRTLIB or RMVM). However, the included text cannot cause declare commands to appear after non-declare commands and cannot contain a PGM(Program) or ENDPGM (End Program) command or another INCLUDE command.

Restrictions:

- The INCLUDE command is valid only within a CL program or ILE CL procedure.
- You must have use (*USE) authority to the file specified for the Source file (SRCFILE) parameter, and execute (*EXECUTE) authority to the library wich contains the source file.

Top

Parameters

Keyword	Description	Choices	Notes
SRCMBR	Source member	*Name*	Required, Positional 1
SRCFILE	Source file	Single values: ***INCFILE** Other values: *Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Source member (SRCMBR)

Specifies the source member that contains the CL program source to be included.

This is a required parameter.

name Specify the name of the source member.

Top

Source file (SRCFILE)

Specifies the source file that contains the source member to be included.

Single Values

***INCFILE**
The file specified for the INCLUDE file (INCFILE) parameter on the CL command used to invoke the CL compiler is used.

Qualifier 1: Source file

name Specify the name of the source file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is used to locate the source file. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Examples

Example 1: Include CL Source Using Default INCLUDE File

```
INCLUDE   SRCMBR(DCLSET1)  SRCFILE(*INCFILE)
```

This command will cause the CL program source in member DCLSET1 to be included at compile time. Member DCLSET1 must be located in the file specified for the INCFILE parameter on the CL command which invoked the CL compiler.

Example 2: INCLUDE Source From Specified File

```
INCLUDE   SRCMBR(SUBR1)  SRCFILE(MYLIB/COMMONSUBR)
```

This command will cause the CL program source in member SUBR1 to be included at compile time. Member SUBR1 must be located in file COMMONSUBR in library MYLIB.

Top

Error messages

None

Top

Install Linux Server (INSLNXSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Install Linux Server (INSLNXSVR) command installs a supported Linux or VMware Operating System on an integrated server. INSLNXSVR also installs Integrated Server Support code on the server.

Linux server installation occurs in two steps. During the first step, the INSLNXSVR command creates necessary objects to manage the server. This includes a network server description, message queue, line descriptions, storage spaces and TCP/IP interfaces.

During the second step of the Linux server installation, the integrated server is varied on to start the Linux server installation.

Further Linux server installation is performed using the integrated server console and the normal Linux server install process.

When INSLNXSVR completes normally, the Linux server is left in a varied on state.

Restrictions:

1. You must have input/output system configuration (*IOSYSCFG), all object (*ALLOBJ) and job control (*JOBCTL) special authorities to run this command.
2. This command may require interactive input using the integrated server console.
3. The integrated server must be varied off initially.
4. The integrated server may reboot during the second step of the install as the Linux server installs.

Usage Notes:

Any errors that occur during the first step of configuring the integrated server will result in the failure of this command.

After this command is run, if you need to manage the different resources created, use the following commands:

- To check out the status of the Linux server, use the Work with Configuration Status command; WRKCFGSTS CFGTYPE(*NWS).
- To manage the server just installed, use the Work with Network Server Descriptions command; WRKNWSD NWSD(nwsdname).
- To manage the line descriptions created by this command, use the Work with Line Descriptions command; WRKLIND LIND(nwsdname*). The line descriptions are named using the network server name (NWSD parameter) specified on the INSLNXSVR command.
- To manage the TCP/IP interfaces created by this command, use the Work with TCP/IP Network Status (NETSTAT) command, option 1. Another option is to use the Configure TCP/IP (CFGTCP) command, option 1.
- To manage the network server configurations just created by this command, use the Work with NWS Configuration command; WRKNWSCFG NWSCFG(nwsdname*). The network server configurations are named using the network server name (NWSD parameter) specified on the INSLNXSVR command.

Top

Parameters

Keyword	Description	Choices	Notes
NWSD	Network server description	*Communications name*	Required, Key, Positional 1
LNXSVRDST	Linux server distribution	*Character value*	Required, Positional 2
TCPPORTCFG	TCP/IP port configuration	Single values: ***NONE** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Port	1, 2, 3, 4	
	Element 2: Linux internet address	*Character value*	
	Element 3: Linux subnet mask	*Character value*	
	Element 4: Linux gateway address	*Character value*, ***NONE**	
VRTETHPORT	Virtual Ethernet port	Single values: ***NONE** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Port	*VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Linux internet address	*Character value*	
	Element 3: Linux subnet mask	*Character value*	
	Element 4: Associated port	*Name*, ***NONE**	
TCPDMNNAME	TCP/IP local domain name	*Character value*, ***SYS**	Optional
TCPNAMSVR	TCP/IP name server system	Single values: ***SYS**, *NONE Other values (up to 3 repetitions): *Character value*	Optional
SVRSTGSIZE	Server storage space sizes	*Element list*	Optional
	Element 1: Install source size	200-2047, ***CALC**	
	Element 2: System size	1024-1024000, ***CALC**	
SVRSTGASP	Storage space ASP	*Element list*	Optional
	Element 1: Install source ASP	1-255, **1**	
	Element 2: System ASP	1-255, **1**	
STGASPDEV	Server storage ASP device	*Element list*	Optional
	Element 1: Install source ASP device	*Name*	
	Element 2: System ASP device	*Name*	
LNGVER	Language version	*Integer*, ***PRIMARY**, 2922, 2923, 2924, 2928, 2929, 2930, 2931, 2938, 2939, 2940, 2956, 2962, 2963, 2966, 2980, 2981, 2984, 2986, 2987, 2989, 2996	Optional
SYNCTIME	Synchronize date and time	***NONE**, *YES, *NO	Optional
RSTDDEVRSC	Restricted device resources	Single values: ***NONE**, *ALL Other values (up to 10 repetitions): *Name*, *ALLTAPE, *ALLOPT	Optional
SHUTDTIMO	Shutdown timeout	2-45, **15**	Optional
ACTTMR	Activation timer	30-1800, **120**	Optional

Keyword	Description	Choices	Notes
CMNMSGQ	Communications message queue	Single values: ***SYSOPR** Other values: *Qualified object name*	Optional
	Qualifier 1: Communications message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
STGPTH	Storage path	*Element list*	Optional
	Element 1: Network server host adapter	*Name*	
POOL	Pool identifier	***BASE**, *SHRPOOL1, *SHRPOOL2, *SHRPOOL3, *SHRPOOL4, *SHRPOOL5, *SHRPOOL6, *SHRPOOL7, *SHRPOOL8, *SHRPOOL9, *SHRPOOL10, *SHRPOOL11, *SHRPOOL12, *SHRPOOL13, *SHRPOOL14, *SHRPOOL15, *SHRPOOL16, *SHRPOOL17, *SHRPOOL18, *SHRPOOL19, *SHRPOOL20, *SHRPOOL21, *SHRPOOL22, *SHRPOOL23, *SHRPOOL24, *SHRPOOL25, *SHRPOOL26, *SHRPOOL27, *SHRPOOL28, *SHRPOOL29, *SHRPOOL30, *SHRPOOL31, *SHRPOOL32, *SHRPOOL33, *SHRPOOL34, *SHRPOOL35, *SHRPOOL36, *SHRPOOL37, *SHRPOOL38, *SHRPOOL39, *SHRPOOL40, *SHRPOOL41, *SHRPOOL42, *SHRPOOL43, *SHRPOOL44, *SHRPOOL45, *SHRPOOL46, *SHRPOOL47, *SHRPOOL48, *SHRPOOL49, *SHRPOOL50, *SHRPOOL51, *SHRPOOL52, *SHRPOOL53, *SHRPOOL54, *SHRPOOL55, *SHRPOOL56, *SHRPOOL57, *SHRPOOL58, *SHRPOOL59, *SHRPOOL60	Optional
VRTETHPTH	Virtual Ethernet path	Values (up to 5 repetitions): *Element list*	Optional
	Element 1: Port	***VRTETHPTP**, *VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Network server host adapter	*Name*	
VRTETHCTLP	Virtual Ethernet control port	1024-65535, **8800**	Optional
RMTNWSCFG	Remote system NWSCFG	*Name*, ***DFT**	Optional
SPNWSCFG	Service processor NWSCFG	*Name*, ***DFT**	Optional
CNNNWSCFG	Connection security NWSCFG	*Name*, ***DFT**	Optional
INZSP	Initialize service processor	***NONE**, *SYNC	Optional
ENBUNICAST	Enable unicast	***YES**, *NO	Optional
EID	Enclosure identifier	Single values: ***AUTO** Other values: *Element list*	Optional
	Element 1: Serial number	*Character value*	
	Element 2: Manufacturer type and model	*Character value*	
SPNAME	Service processor name	*Character value*, ***SPINTNETA**	Optional
SPINTNETA	SP internet address	*Character value*	Optional
SPAUT	SP authentication	Single values: *DFT Other values: *Element list*	Optional
	Element 1: User name	*Character value*	
	Element 2: User password	*Character value*	

Keyword	Description	Choices	Notes
RMTSYSID	Remote system identifier	Single values: ***EID** Other values: *Element list*	Optional
	Element 1: Serial number	*Character value*	
	Element 2: Manufacturer type and model	*Character value*	
DELIVERY	Delivery method	***DYNAMIC**, *MANUAL	Optional
CHAPAUT	Target CHAP authentication	Single values: *NONE Other values: *Element list*	Optional
	Element 1: CHAP name	*Character value*, ***NWSCFG**	
	Element 2: CHAP secret	*Character value*, ***GEN**	
INRCHAPAUT	Initiator CHAP authentication	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: CHAP name	*Character value*, ***NWSCFG**	
	Element 2: CHAP secret	*Character value*, ***GEN**	
BOOTDEVID	Boot device ID	Single values: ***SINGLE** Other values: *Element list*	Optional
	Element 1: Bus	0-255	
	Element 2: Device	0-31	
	Element 3: Function	0-7	
DYNBOOTOPT	Dynamic boot options	*Element list*	Optional
	Element 1: Vendor ID	*Character value*, ***DFT**	
	Element 2: Alternate client ID	*Character value*, ***ADPT**	
RMTIFC	Remote (initiator) interfaces	*Element list*	Optional
	Element 1: SCSI interface	*Element list*	
	Element 1: Adapter address	*Hexadecimal value*	
	Element 2: Internet address	*Character value*	
	Element 3: Subnet mask	*Character value*	
	Element 4: Gateway address	*Character value*, ***NONE**	
	Element 5: iSCSI qualified name	*Character value*, ***GEN**	
	Element 2: LAN interface	*Element list*	
	Element 1: Adapter address	*Hexadecimal value*	
	Element 2: Internet address	*Character value*	
	Element 3: Subnet mask	*Character value*	
	Element 4: Gateway address	*Character value*, ***NONE**	
TEXT	Text ’description’	*Character value*, ***BLANK**	Optional
CFGFILE	Configuration file	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Configuration file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
VRTPTPPORT	Virtual PTP Ethernet port	*Element list*	Optional
	Element 1: Internet address	*Character value*, ***GEN**	
	Element 2: Linux internet address	*Character value*, ***GEN**	

Top

Network server description (NWSD)

Specifies the name of the network server to be installed. The network server description is created using the values specified in this command. The Create Network Server Description (CRTNWSD) command will be used to create the NWSD. The name should also used as the computer name of the integrated server that is installed as well as the TCP host name for the integrated server.

This is a required parameter.

communications-name
Specify the name of the network server description.

The network server name can be up to eight characters. The following characters are allowed in NWSD names:

- Alphabetical characters A through Z
- Digits 0 through 9

Top

Linux server distribution (LNXSVRDST)

Specifies the integrated server operating system version to install on the integrated server. Use F4 while prompting the command to see the complete list of allowed values for this parameter.

This is a required parameter.

Top

TCP/IP port configuration (TCPPORTCFG)

Specifies the TCP/IP configuration values that are specific to a port on the integrated server.

This information consists of four parts including the identification of the integrated server port, the internet address, the subnet mask and the default gateway assigned to the port.

Single values

***NONE**
There is no TCP/IP port configuration.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the locally managed integrated server port number.

1 Integrated server port number 1 is configured.

2 Integrated server port number 2 is configured.

3 Integrated server port number 3 is configured.

4 Integrated server port number 4 is configured.

Element 2: Linux internet address

character-value
Specify the internet address associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

An internet address that has a binary value of all ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the internet address is not valid.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 3: Linux subnet mask

character-value
Specify the subnet mask associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Linux gateway address

***NONE**
There is no gateway address.

character-value
Specify the gateway address associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Virtual Ethernet port (VRTETHPORT)

Specifies the TCP/IP configuration for the virtual Ethernet used by the integrated server.

Note: The VRTETHPORT parameter is only available for Integrated xSeries servers.

Single values

***NONE**
Do not configure any virtual Ethernet ports.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the virtual Ethernet port number.

***VRTETHn**
The network server virtual Ethernet port ’n’ is configured, where ’n’ can have a value of 0 through 9.

Note: Each value can only be specified once.

Element 2: Linux internet address

character-value
Specify the internet address associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

An internet address that has a binary value of all ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the internet address is not valid.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 3: Linux subnet mask

character-value
Specify the subnet mask associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Associated port

Specifies the resource name that describes the port that is used to establish a connection between the integrated server and the network.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name. The resource name is on the port. For example, the resource name may be CMN01 on a Ethernet port.

***NONE**
An associated port resource name is not associated with the line.

name Specify the associated port resource name.

Top

TCP/IP local domain name (TCPDMNNAME)

Specifies the local domain name associated with the integrated server.

A domain name can be a text string having 1 to 255 characters. Domain names consist of one or more labels separated by periods. Each label can contain up to 63 characters. The following characters are allowed in domain names:

- Alphabetical characters A through Z
- Digits 0 through 9
- Minus sign (-)
- Underscore (_)
- Period (.). Periods are only allowed when they separate labels of domain style name (refer to RFC 1034).

Uppercase and lowercase characters are allowed, but no significance attached to the case. The case is maintained as entered. The first and last character of the host name must be an alphabetic character or a digit.

***SYS** Specifies that the local domain name for the integrated server should be the same value as is configured for i5/OS.

character-value
Specify a TCP domain name to be associated with the integrated server.

Top

TCP/IP name server system (TCPNAMSVR)

Specifies the internet address of the name server system that is used by the integrated server. Typically, this is the same value as it is for i5/OS.

Single values

***SYS** The name server systems used by the integrated server should be the same as for i5/OS.

***NONE**
There is no name server to be used by the integrated server.

Other values (up to 3 repetitions)

character-value
Specify an internet address for the name server system to be used by the integrated server. Up to three remote name server systems can be specified. The name server systems are used in the order they are specified.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Server storage space sizes (SVRSTGSIZE)

Specifies the size of the server storage spaces, in megabytes.

Notes:

1. The contents of the DOSUTILS directory of the Linux server installation media and the Integrated Server Support code are copied to the install source drive. The size specified for the install source drive must be large enough to hold this data.

Element 1: Install source size

Specifies the size of the storage space that holds the files used to install the integrated server operating system.

***CALC**
Specify that the system should calculate the size based on the space required to install the integrated server.

200-2047
Specify the install source size value in megabytes.

Element 2: System size

Specifies the size in megabytes of the storage space that the integrated server operating system is installed on.

***CALC**
Specify that the size should be calculated by the system based on the values specified on other parameters.

1024-1024000
Specify the system size value in megabytes.

Top

Storage space ASP (SVRSTGASP)

Specifies the auxiliary storage pool (ASP) identifiers for the storage space that will contain the files used to install the integrated server and the storage space that will contain the integrated server operating system.

Note: You cannot specify both a SVRSTGASP and STGASPDEV parameter value for the same element.

Note: Null (omitted) values are specified with the characters *N, which mean that no value was specified. The value specified for the corresponding Server storage ASP device (STGASPDEV) element will be used if specified. Otherwise, the default value will be used. *N is needed only when another value following the omitted element is being specified.

Element 1: Install source ASP

Specifies the auxiliary storage pool for the storage space that holds the files that are used to install the integrated server operating system.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

2-255 Specify the number of the ASP to be used. Valid values depend on what ASPs are defined on the system.

Element 2: System ASP

Specifies the auxiliary storage pool for the storage space that holds the integrated server operating system.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

2-255 Specify the number of the ASP to be used. Valid values depend on what ASPs are defined on the system.

Top

Server storage ASP device (STGASPDEV)

Specifies the auxiliary storage pool (ASP) device name for the storage space that will contain the files used to install the integrated server operating system and the storage space that will contain the integrated server operating system.

Note: You cannot specify both a SVRSTGASP and STGASPDEV parameter value for the same element.

Note: The ASP must have been activated (by varying on the ASP device) and have a status of ’Available’.

Element 1: Install source ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the files used to install the integrated server operating system.

name The device name of the ASP to use for the network server storage space.

Element 2: System ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the integrated server operating system.

name The device name of the ASP to use for the network server storage space.

Top

Language version (LNGVER)

Specifies the installed language environment used to display Integrated Server Support text and messages. Note that a smaller set of languages are available for messages displayed during installation than for messages displayed after installation.

***PRIMARY**
The installed language environment for Integrated Server Support text and messages is based on the language feature of the system primary language.

integer
Specify the language feature that will be used to select the Integrated Server Support text and messages.

Top

Synchronize date and time (SYNCTIME)

Specifies whether i5/OS should synchronize the integrated server date and time with i5/OS date and time.

The QTIMZON system value must be set to the correct value for time synchronization to work correctly.

***NONE**
i5/OS will never synchronize the integrated server date and time with the i5/OS date and time when the network server description is varied on.

***YES** i5/OS will synchronize integrated server date and time with i5/OS date and time.

***NO** i5/OS only synchronizes the integrated server date and time with i5/OS date and time once when the network server description is varied on.

Top

Restricted device resources (RSTDDEVRSC)

Specifies the tape and optical device resource names that are restricted and cannot be used by the integrated server.

The resource is used when the integrated server is active and a request is issued from a client that is running an application. The device resource cannot be used by the client application and i5/OS based application at the same time. If the device resource is intended to be used by an application, it will need to be available when the application is ready to use it.

Single values

***NONE**
No device resources are restricted from the integrated server. Therefore, any tape or optical device resources that exist on the system can be used.

***ALL** All tape and optical device resources are restricted from being used by the integrated server.

Other values (up to 10 repetitions)

***ALLOPT**
All optical device resources are restricted from being used by the integrated server.

Note: This value can only be specified once.

***ALLTAPE**
All tape resources are restricted from being used by the integrated server.

Note: This value can only be specified once.

name Specify the resource name of a restricted device that cannot be used by the integrated server. Up to 10 restricted device resource names can be specified.

Top

Shutdown timeout (SHUTDTIMO)

Specifies the shutdown timeout value in minutes. This is used to limit the amount of time that the integrated server operating system is allowed to shutdown before the network server description is varied off.

15 The integrated server default shutdown time-out value is used.

2-45 Specify the number of minutes to wait. i5/OS waits until the integrated server operating system has shutdown successfully, or until the specified time passes before varying the network server description off.

Top

Activation timer (ACTTMR)

Specifies the amount of time (in seconds) the system will wait for the connection to be established to the remote server service processor and to power on the remote server.

120 The activate time of 120 seconds is used.

integer
Specify, in seconds, a value ranging from 30 through 1800.

Top

Communications message queue (CMNMSGQ)

Specifies the name of a message queue to receive network server host adapter communications status messages.

Single values

***SYSOPR**
Causes network server host adapter communications status messages to be placed in the system operator message queue.

Qualifier 1: Communications message queue

name Specify the name of a message queue to receive network server host adapter communications status messages.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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.

name Specify the name of the library to be searched.

Top

Storage path (STGPTH)

Specifies the storage path the storage spaces can use.

Element 1: Network server host adapter

name Specify the name of an existing **Network server host adapter** (NWSH) device description.

Top

Pool identifier (POOL)

Specifies the shared data storage pool this integrated server should use.

***BASE**
The base pool is to be used by this integrated server.

***SHRPOOLnn**
Specifies the shared pool to be used by this integrated server. There are sixty general-purpose shared pools, identified by special values *SHRPOOL1 to *SHRPOOL60.

Top

Virtual Ethernet path (VRTETHPTH)

Specifies the virtual Ethernet paths the Ethernet line descriptions can use. This information consists of two parts including the virtual Ethernet port and the **Network server host adapter** (NWSH) description. You must enter at least one virtual Ethernet path which is the path to be used by the *VRTETHPTP line description.

You can specify 5 values for this parameter.

Element 1: Port

***VRTETHPTP**
The network server virtual Ethernet point to point port is configured.

***VRTETHn**
The virtual Ethernet port ′n′ is configured, where ′n′ can have a value of 0 through 9.

Element 2: Network server host adapter

name Specify the name of an existing **Network server host adapter** (NWSH) description. The network server host adapter name does not need to be unique for each VRTETHPTH parameter on this NWSD.

Top

Virtual Ethernet control port (VRTETHCTLP)

Specifies the TCP port to use for virtual Ethernet control.

8800 Use the TCP port number of 8800.

1024-65535
Specify the port number identifying the port that is to be used for virtual Ethernet control.

Top

Remote system NWSCFG (RMTNWSCFG)

Specifies the remote system network server configuration to use with this server.

***DFT** Use the system generated default remote system network server configuration name of ′nwsdnameRM′ where nwsdname is the name of the network server description.

name Specify the name of an existing remote system network server configuration.

Top

Service processor NWSCFG (SPNWSCFG)

Specifies the service processor network server configuration to use with this server.

***DFT** Use the system generated default service processor network server configuration name of ′nwsdnameSP′ where nwsdname is the name of the network server description.

name Specify the name of an existing service processor network server configuration.

Top

Connection security NWSCFG (CNNNWSCFG)

Specifies the connection security network server configuration to use with this server.

***DFT** Use the system generated default connection security network server configuration name of ′nwsdnameCN′ where nwsdname is the name of the network server description.

name Specify the name of an existing connection security network server configuration.

Top

Initialize service processor (INZSP)

Specifies how the remote system service processor is secured.

***NONE**

Provides no security.

Use this only if the interconnecting network is physically secure.

Note: Some service processors do not support secure connections. Use *NONE for these service processors. Additional information can be found at System i integration with BladeCenter and System x at http://www.ibm.com/systems/i/bladecenter/.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***SYNC**

Synchronize the user name and password and self-signed certificate from the service processor. This option is used to initially synchronize i5/OS with the service processor. It is also used if multiple service processor network server configurations are used for the same system or the service processor network server configuration has been restored from backup and the service processor certificate must be synchronized. The current user name and password for the service processor must be specified on the **SP authentication (SPAUT)** parameter to perform this option.

Top

Enable unicast (ENBUNICAST)

Specifies whether unicast packet distribution is to be used. Unicast is a transmission method where packets are sent directly to the specified **Service processor name (SPNAME)** or **SP internet address (SPINTNETA)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***YES** Enable unicast.

***NO** Disable unicast

Top

Enclosure identifier (EID)

Specifies the identifying serial number, type and model of the enclosure containing the service processor.

When specified, they are used to locate the system on the network.

Look for these values on the label of the system.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

***AUTO**
Automatically retrieve the identifier when ENBUNICAST(*YES) is specified.

Element 1: Serial number

character-value
Specify the machine serial number.

Element 2: Manufacturer type and model

character-value
Specify the machine type and model.

The value is entered in the form ttttmmm where tttt is the machine type and mmm is the machine model number.

Top

Service processor name (SPNAME)

Specifies the remote system service processor host name.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Notes:

1. This parameter is required when ENBUNICAST(*YES) is specified.
2. This parameter is ignored when ENBUNICAST(*NO) is specified.

***SPINTNETA**
The remote system is identified by the value specified for the **SP internet address (SPINTNETA)** parameter.

character-value
Specify the remote system service processor host name.

Top

SP internet address (SPINTNETA)

Specifies the remote system service processor internet address.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Notes:

1. This parameter is ignored when ENBUNICAST(*NO) is specified.
2. This parameter is required when SPNAME(*SPINTNETA) is specified.

character-value
Specify the internet address of the service processor.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

SP authentication (SPAUT)

Specifies the service processor user name and password. This is used to authenticate and secure the service processor.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

***DFT** The default service processor userid and password are used.

Element 1: User name

character-value
Specify a name that represents the host configuration that owns the service processor. It is suggested that the remote system network server configuration name be used. If multiple remote

system network server configurations use the same service processor at different times, each configuration must contain the same user name and password.

Element 2: User password

character-value
Specify the service processor password. Password must be at least 5 characters in length and contain at least one alphabetic character and one numeric or symbolic character.

Top

Remote system identifier (RMTSYSID)

Specifies the identifying serial number, type and model of the remote system. When specified, they are used to locate the remote system on the network.

Look for these values on the label of the system.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***EID** Use the service processor identifier.

Element 1: Serial number

character-value
Specify the machine serial number.

Element 2: Manufacturer type and model

character-value
Specify the machine type and model.

The value is entered in the form ttttmmm where tttt is the machine type and mmm is the machine model number.

Top

Delivery method (DELIVERY)

Specifies how the parameters necessary to configure the remote system are delivered.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

***DYNAMIC**
Parameters are dynamically delivered to the remote system using Dynamic Host Configuration Protocol (DHCP).

***MANUAL**
Parameters are manually configured on the remote system using the BIOS utilities (System BIOS or Adapter BIOS - CTRL-Q).

Top

Target CHAP authentication (CHAPAUT)

Specifies the Challenge Handshake Authentication Protocol (CHAP) for the System i iSCSI target to authenticate the remote system iSCSI initiators.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***NONE**
CHAP authentication is not enabled.

Element 1: CHAP name

***NWSCFG**
The system will automatically generate a name for CHAP using the Network server configuration name.

character-value
Specify the name you want to use for the Challenge Handshake Authentication Protocol.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Element 2: CHAP secret

***GEN** The system will automatically generate a random CHAP secret.

character-value
Specify the secret you want to use for the Challenge Handshake Authentication Protocol.

Note: Target and initiator CHAP secrets must not be the same.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma

- Underline
- Minus sign
- Period
- Colon
- Semicolon

Top

Initiator CHAP authentication (INRCHAPAUT)

Specifies the Challenge Handshake Authentication Protocol (CHAP) for the remote system iSCSI initiators to authenticate the System i iSCSI target.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***NONE**
CHAP authentication is not enabled.

Element 1: CHAP name

***NWSCFG**
The system will automatically generate a name for CHAP using the Network server configuration name.

character-value
Specify the name you want to use for the Challenge Handshake Authentication Protocol.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Element 2: CHAP secret

***GEN** The system will automatically generate a random CHAP secret.

character-value
Specify the secret you want to use for the Challenge Handshake Authentication Protocol.

Note: Target and initiator CHAP secrets must not be the same.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Top

Boot device ID (BOOTDEVID)

Specifies the PCI Function Address (Bus/Device/Function) of the iSCSI initiator port in the remote system that will be used to boot from.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***SINGLE**
The single iSCSI initiator port is used on the remote system

Element 1: Bus

0-255 Specify the bus number of the remote system iSCSI initiator port that will be used to boot.

Element 2: Device

0-31 Specify the device number of the remote system iSCSI initiator port that will be used to boot.

Element 3: Function

0-7 Specify the function number of the remote system iSCSI initiator port that will be used to boot.

Top

Dynamic boot options (DYNBOOTOPT)

Specifies the internal Dynamic Host Configuration Protocol (DHCP) Server configuration.

Note: This is an advanced configuration function.

This parameter is used to configure the internal DHCP Server that is part of the iSCSI target host bus adapter (HBA) firmware. It is used to provide IP address and diskless boot parameters for the remote system iSCSI initiator port.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

This parameter is only valid when DELIVERY(*DYNAMIC) is specified.

Element 1: Vendor ID

The client and server are pre-configured to a default vendor ID. Network administrators can configure clients to define their own identifying values to convey hardware, operating system or other identifying information. DHCP option 60 described in the IETF RFC 2132 is used for this function.

***DFT** The default vendor ID will be used.

character-value
Vendor ID of the remote system iSCSI initiator port that will be used.

Element 2: Alternate client ID

Used by clients to specify their unique identifier to the server. Each client identifier must be unique among all other client identifiers used on the effective DHCP network to which the client is attached (that is, the client local subnet and any remote subnets reachable using DHCP relay). Vendors and system administrators are responsible for choosing client identifiers that meet this requirement for uniqueness. DHCP option 61 described in the IETF RFC 2132 is used for this function.

***ADPT**
The default Client ID consists of the adapter address for the remote system iSCSI initiator port. This value will be used to identify the remote system.

character-value
Specify the Client ID of the remote system iSCSI initiator port that will be used to boot.

Top

Remote (initiator) interfaces (RMTIFC)

Specifies the remote system iSCSI initiator host bus adapter (HBA) configuration. Each iSCSI initiator port has two functions to support a SCSI and a LAN or TCP Offload Engine (TOE) interface.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Element 1: SCSI interface

Specifies the SCSI interface.

Element 1: Adapter address

hexadecimal-value
Specify the 12-character hexadecimal adapter address for the initiator port.

Element 2: Internet address

character-value
Specify the internet address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

character-value
Specify the subnet mask for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

***NONE**
No gateway address is configured for this initiator port.

character-value
Specify the gateway address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 5: iSCSI qualified name

***GEN** The system will automatically generate the iSCSI qualified name.

character-value
Specify the iSCSI qualified name for the initiator port.

The following characters are allowed in iSCSI qualified names:

- Alphabetical characters A through Z converted to lower case (refer to RFC 3722)
- Alphabetical characters a through z
- Digits 0 through 9
- Period (.)
- Dash (-)
- colon (:)

Element 2: LAN interface

Specifies the LAN interface.

Element 1: Adapter address

hexadecimal-value
Specify the 12-character hexadecimal adapter address for the initiator port.

Element 2: Internet address

character-value
Specify the internet address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

character-value
Specify the subnet mask for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

***NONE**
No gateway address is configured for this initiator port.

character-value
Specify the gateway address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Text ’description’ (TEXT)

Specifies text that briefly describes the network server configuration.

***BLANK**
Text is not specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Configuration file (CFGFILE)

Specifies the source file containing configuration data to be used in activating or further defining the integrated server.

Single values

***NONE**
No configuration file is specified.

Qualifier 1: Configuration file

name Specify the name of the source file containing the configuration data members for the server. At the time the server is activated, all members in the file are processed. The file must exist on the system by the time the server is activated.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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.

name Specify the name of the library to be searched.

Top

Virtual PTP Ethernet port (VRTPTPPORT)

Specifies the TCP/IP configuration for the virtual point-to-point Ethernet port.

Notes:

1. The VRTPTPPORT parameter must be used for integrated servers running on the host system.

2. The subnet mask that is used for both sides of the virtual point-to-point Ethernet port is 255.255.255.0 by default. Therefore, the internet addresses that are chosen for both sides of the virtual point-to-point Ethernet port must have the same values for the first three parts of the internet addresses.

Element 1: Internet address

Specifies the internet address for the i5/OS side of the virtual point-to-point Ethernet connection.

***GEN** Specify *GEN to let the system configure a virtual point-to-point Ethernet port with a generated internet address.

character-value
Specify the i5/OS internet address for the virtual point-to-point Ethernet port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 2: Linux internet address

Specifies the internet address for the integrated server side of the virtual point-to-point Ethernet connection.

***GEN** Specify *GEN to let the system configure a virtual point-to-point Ethernet port with a generated internet address.

character-value
Specify the integrated server internet address for the virtual point-to-point Ethernet port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Top

Examples

Example 1: Install Linux Using iSCSI Hardware

```
INSLNXSVR   NWSD(RHEL5MAR)  LNXSVRDST(*RHEL5)
            SVRSTGSIZE(200 12000)  SVRSTGASP(2 2)  LNGVER(2920)
            SHUTDTIMO(2)  ACTTMR(300)  ENBUNICAST(*YES)
            SPINTNETA('9.5.2.32')  SPAUT(JSMITH ())
            TEXT('RedHat 5 server')
```

This command will install a Linux server with a Red Hat 5 distribution on iSCSI hardware. It will contain a 12 GB system drive which resides on user ASP 2. The name RHEL5MAR will be associated with the network server description on the i5/OS side. The language version used will be 2920. The shut down timeout will be set to 2 minutes. The system will wait 300 seconds for the connection to be made to the remote server service processor. Unicast packet distribution will be used over iSCSI. The remote server service processor internet address will be set to 9.5.2.32. SP authentication will be made with username JSMITH.

Top

Error messages

***ESCAPE Messages**

NTA1007
Network server description &1 must be varied off.

NTA1199
Vary on or off of the Linux server not successful.

NTA119A
Linux server installation not successful.

NTA1024
Storage space assigned to network server &1 missing, damaged or not valid.

NTA1030
Internal error occurred.

Top

Install Program Temporary Fix (INSPTF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Install Program Temporary Fix (INSPTF) command allows the user to load and apply PTFs for multiple products with a single command. PTF groups will be copied to the system when they do not already exist on the system or when the level of the PTF group on the media is higher than the level of the PTF group that exists on the system. If LICPGM(*ALL) is specified, any special handling PTFs listed in PTF groups will be used during the install.

The OMIT and HIPER parameters are supplied to allow the user of the INSPTF command to be more selective. These parameters apply only to the PTF loading activity. Any PTF already loaded on the system will be applied.

The INSTYP parameter controls the apply of the PTFs. Using the different special values allows immediate and delayed apply combinations as well as starting an IPL.

INSPTF does not support loading PTFs from tape for products that have multiple releases of the base option installed on the system. If PTFs for such a product exist on the tape, the INSPTF will not load those PTFs and will return an error.

Top

Parameters

Keyword	Description	Choices	Notes
LICPGM	Product description	Values (up to 300 repetitions): *Element list*	Required, Positional 1
	Element 1: Product	*Character value*, *ALL	
	Element 2: Release	*Character value*, ***ONLY**	
DEV	Device	*Name*, ***SERVICE**, *NONE	Optional
INSTYP	PTF apply type	***SRVATT**, *DLYIPL, *DLYALL, *IMMONLY, *IMMDLY	Optional
OMIT	PTF omit list	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Product	*Character value*	
	Element 2: PTF identifier	*Character value*	
	Element 3: Release	*Character value*, ***ONLY**	
HIPER	HIPER PTFs only	*YES, ***NO**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
RESTART	Restart type	***IPLA**, *SYS, *FULL	Optional
PMTMED	Prompt for media	***SNGVOLSET**, *MLTVOLSET, *MLTSRV	Optional
CPYPTF	Copy PTFs	***SRVATT**, *YES, *NO	Optional

Top

Product description (LICPGM)

Specifies the product ID, version, release, and modification level of the products for which PTFs should be installed.

This is a required parameter.

You can specify 300 values for this parameter.

Element 1: Product

***ALL** The available PTFs for all installed products are installed. This must be the first and only value if specified. All values specified after it are ignored.

character-value
Specify the product ID of the PTFs to be installed.

Element 2: Release

***ONLY**
This value is valid only when one release of the product's base option is installed on the system. PTFs for all installed options of the product are loaded and applied regardless of the release-level of the option.

character-value
Specify the release level of the base product option in the format VxRyMz, where Vx is the version number, Ry is the release number, and Mz is the modification level.

Top

Device (DEV)

Specifies the device from which the PTFs are loaded. The device name must already be known on the system by a device description.

***SERVICE**
The PTFs sent from the service support system are installed.

***NONE**
No PTFs are loaded. Any PTF already loaded on the system will be applied. This special value is used after the IPL following an incomplete PTF install. Special handling PTFs in a PTF group being installed must be applied and active before the remaining PTFs in the PTF group can be applied.

name Specify the name of the tape device or optical device from which the PTFs are installed.

Top

PTF apply type (INSTYP)

Specifies the type of install to perform.

***SRVATT**
The type of install depends on the service attribute setting.

Attention:

The service attribute is shipped with *DLYIPL as the default. Use the Change Service Attributes (CHGSRVA) command to change the default.

***DLYIPL**
All PTFs are marked for delayed apply and an initial program load (IPL) is done on the system.

***DLYALL**
All PTFs are marked for delayed apply and an initial program load (IPL) is not done on the system.

***IMMDLY**
The immediate PTFs are applied and the delayed PTFs are marked for apply at the next initial program load (IPL).

***IMMONLY**
All PTFs are loaded, but only the immediate PTFs are applied and an initial program load (IPL) is not done on the system.

Top

PTF omit list (OMIT)

Specifies the product ID, version, release, modification level, and PTF ID for PTFs that should not be loaded. The current state of the PTF is not checked before being passed to LODPTF. If the PTF is already loaded it is applied. A maximum of 50 PTFs can be omitted.

Element 1: Product

character-value
Specify the product ID for the PTFs that should not be loaded.

Element 2: PTF identifier

character-value
Specify the PTF ID for the PTFs that should not be installed.

Element 3: Release

***ONLY**
The only release of the product selected on the LICPGM parameter.

character-value
Specify the release level of the base product option or the release level of the PTF for the PTFs that should not be loaded. The release level must be specified in VxRyMz format, where Vx is the version number, Ry is the release number, and Mz is the modification level.

Top

HIPER PTFs only (HIPER)

Specifies whether only high-impact pervasive (HIPER) PTFs should be loaded when installing from a media.

Note: This parameter is ignored when *SERVICE is specified for the **Device (DEV)** parameter. This is valid only when installing IBM cumulative PTF packages.

***NO** All PTFs, other than those listed in the omit list, should be installed.

***YES** Only HIPER PTFs that are not on the omit list should be installed.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically performed on the tape or optical volume after the PTF operation ends.

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

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

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

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

Top

Restart type (RESTART)

Specifies the point from which the initial program load (IPL) restarts when the PTF apply type (INSTYP) parameter indicates an IPL will be performed.

Note: This is valid only when INSTYP(*DLYIPL) is specified or when INSTYP(*SRVATT) is specified and the PTF install type (PTFINSTYP) service attribute is set to *DLYIPL.

<u>*IPLA</u> The value specified on the Change IPL Attributes (CHGIPLA) command is used. To determine the current setting for this value, use the Display IPL Attributes (DSPIPLA) command.

***SYS** Specifies that the system determines how much of the system to restart.

The operating system is always restarted. The hardware is restarted only if a PTF that requires a restart is applied. Other hardware functions, such as some configuration changes, that occur during a *FULL IPL are not processed.

*SYS can result in a shorter IPL time than if you specify *FULL.

***FULL** All portions of the system, including the hardware, are restarted.

Top

Prompt for media (PMTMED)

Specifies whether the user will be prompted for additional PTF volume sets and load PTFs from *SERVICE after loading PTFs from a device.

Note: This parameter is ignored when *SERVICE or *NONE is specified for the **Device (DEV)** parameter.

<u>*SNGVOLSET</u>
The user will be prompted to mount each volume in a single volume set when loading the PTFs. If a virtual optical device is specified on the DEV parameter, all mounted PTF volumes will be processed.

***MLTVOLSET**
The user will be prompted for volumes in multiple volume sets when loading the PTFs.

***MLTSRV**
The user will be prompted for volumes in multiple volume sets when loading the PTFs. After PTFs are loaded from the last volume set, PTFs will be loaded from the service support system (*SERVICE).

Top

Copy PTFs (CPYPTF)

Specifies whether to copy PTF save files and cover letters into *SERVICE when PTFs are loaded. PTF save files must be in *SERVICE when distributing PTFs to other systems or when using the Save System Information (SAVSYSINF) command.

Note: This parameter is ignored if DEV(*SERVICE) or DEV(*NONE) is specified.

***SRVATT**
Use the Copy PTFs (CPYPTF) service attribute to determine if PTF save files and cover letters will be copied into *SERVICE when PTFs are loaded. The Display Service Attributes (DSPSRVA) command displays information about how the system is set up. This includes whether PTF save files and cover letters will be copied into *SERVICE when PTFs are loaded. The Change Service Attributes (CHGSRVA) command can be used to change the CPYPTF service attribute.

***YES** PTF save files and cover letters that do not already exist are copied into *SERVICE when PTFs are loaded.

***NO** PTF save files and cover letters are not copied into *SERVICE when PTFs are loaded.

Top

Examples

Example 1: Omitting PTFs

```
INSPTF   LICPGM((*ALL))  DEV(*SERVICE)  INSTYP(*IMMDLY)
         OMIT((5761999 MF12345 V5R4M0) (5761SS1 SI12345 V5R4M0))
```

This command will load all PTFs that are in *SERVICE for all products installed on the system except MF12345 and SI12345. It will then apply all PTFs in loaded status that can be applied immediately and mark the rest for delayed apply.

Example 2: Installing HIPER only

```
INSPTF   LICPGM((5761PT1 V5R4M0))  DEV(TAP01)  INSTYP(*IMMONLY)
         HIPER(*YES)
```

This command will search the media for PTFs for the V5R4M0 Performance Tools product in the HIPER section. Each PTF that can be applied immediately will be. Delayed PTFs will be loaded, but not marked for apply.

Example 3: Installing Only Immediate PTFs

```
INSPTF   LICPGM((*ALL))  DEV(TAP01)  INSTYP(*IMMONLY)
         ENDOPT(*LEAVE)
```

This command will load all PTFs for the products that are installed on the system, from the device TAP01. All PTFs in loaded status on the system that can be applied immediately will be. No delayed PTFs will be set for apply.

Top

Error messages

***ESCAPE Messages**

CPF358A
Release not valid.

CPF358F
LICPGM parameter contains duplicate entries.

CPF35EB
Multiple releases of product &1 installed.

CPF3586
List of PTFs not correct.

CPF36B7
PTF install processing incomplete; IPL required.

CPF3606
Product &1 &2 not installed.

CPF361A
PTFs installed successfully, but actions pending.

CPF361B
PTF install processing failed, and there are actions pending.

CPF361C
No PTFs installed.

CPF3615
PTF install processing failed.

CPF3618
The mode is not set at Normal.

Top

Install Windows Server (INSWNTSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Install Windows Server (INSWNTSVR) command installs the Windows server Operating System on an integrated server. INSWNTSVR also installs Integrated Server Support code on the server.

Windows server installation occurs in two steps. During the first step, the INSWNTSVR command creates necessary objects to manage the server. This includes a network server description, message queue, line descriptions, storage spaces and TCP/IP interfaces.

During the second step of the Windows server installation, the integrated server is varied on to start the Windows server installation.

For INSTYPE of *BASIC, the ServerGuide CD is inserted into the servers locally attached CD-ROM or DVD drive and the server is varied on. At this point, the INSWNTSVR command ends. The ServerGuide CD in the server is booted from and continues with the process of configuring the server, including detecting and configuring any devices or adapters and preparing the system for the Windows installation.

Further Windows server installation is performed using the integrated server console and the normal Windows server install process.

When INSWNTSVR completes normally, the Windows server is left in a varied on state.

Restrictions:

1. You must have input/output system configuration (*IOSYSCFG), all object (*ALLOBJ) and job control (*JOBCTL) special authorities to run this command.
2. This command may require interactive input using the integrated server console.
3. The integrated server must be varied off initially.
4. The integrated server may reboot during the second step of the install as the Windows server installs.

Usage Notes:

Any errors that occur during the first step of configuring the integrated server will result in the failure of this command.

After this command is run, if you need to manage the different resources created, use the following commands:

- To check out the status of the Windows server, use the Work with Configuration Status command; WRKCFGSTS CFGTYPE(*NWS).
- To manage the server just installed, use the Work with Network Server Descriptions command; WRKNWSD NWSD(nwsdname).
- To manage the line descriptions created by this command, use the Work with Line Descriptions command; WRKLIND LIND(nwsdname*). The line descriptions are named using the network server name (NWSD parameter) specified on the INSWNTSVR command.
- To manage the TCP/IP interfaces created by this command, use the Work with TCP/IP Network Status (NETSTAT) command, option 1. Another option is to use the Configure TCP/IP (CFGTCP) command, option 1.

- To manage the network server configurations just created by this command, use the Work with NWS Configuration command; WRKNWSCFG NWSCFG(nwsdname*). The network server configurations are named using the network server name (NWSD parameter) specified on the INSWNTSVR command.

Top

Parameters

Keyword	Description	Choices	Notes
NWSD	Network server description	*Communications name*	Required, Key, Positional 1
INSTYPE	Installation type	*FULL, *BASIC	Required, Positional 2
RSRCNAME	Resource name	*Name*, *ISCSI	Required, Positional 3
WNTVER	Windows server version	*WIN2000, *WIN2003, *WIN2008	Required, Positional 4
WNTSRCDIR	Windows source directory	*Path name*, ***DFT**	Optional
OPTION	Install option	***INSTALL**, *UPGRADE	Optional
TCPPORTCFG	TCP/IP port configuration	Single values: ***NONE** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Port	1, 2, 3, 4	
	Element 2: Windows internet address	*Character value*	
	Element 3: Windows subnet mask	*Character value*	
	Element 4: Windows gateway address	*Character value*, ***NONE**	
VRTETHPORT	Virtual Ethernet port	Single values: ***NONE** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Port	*VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Windows internet address	*Character value*	
	Element 3: Windows subnet mask	*Character value*	
	Element 4: Associated port	*Name*, ***NONE**	
TCPDMNNAME	TCP/IP local domain name	*Character value*, ***SYS**	Optional
TCPNAMSVR	TCP/IP name server system	Single values: ***SYS**, *NONE Other values (up to 3 repetitions): *Character value*	Optional
MSGQ	Server message queue	Single values: ***JOBLOG**, *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Server message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
EVTLOG	Event log	Single values: ***ALL**, *NONE Other values (up to 3 repetitions): *SYS, *SEC, *APP	Optional
SVRSTGSIZE	Server storage space sizes	*Element list*	Optional
	Element 1: Install source size	200-2047, ***CALC**	
	Element 2: System size	1024-1024000, ***CALC**	

Keyword	Description	Choices	Notes
SVRSTGASP	Storage space ASP	*Element list*	Optional
	Element 1: Install source ASP	1-255, **1**	
	Element 2: System ASP	1-255, **1**	
STGASPDEV	Server storage ASP device	*Element list*	Optional
	Element 1: Install source ASP device	*Name*	
	Element 2: System ASP device	*Name*	
CVTNTFS	Convert to NTFS	***YES**, *NO	Optional
TOWRKGRP	To workgroup	*Character value*	Optional
TODMN	To domain	*Character value*	Optional
FULNAM	Full Name	*Character value*	Optional
ORG	Organization	*Character value*	Optional
LNGVER	Language version	*Integer*, ***PRIMARY**, 2911, 2922, 2923, 2924, 2925, 2926, 2928, 2929, 2931, 2932, 2933, 2937, 2938, 2939, 2940, 2942, 2950, 2962, 2963, 2966, 2975, 2976, 2978, 2979, 2980, 2981, 2984, 2986, 2987, 2989, 2994, 2996	Optional
SYNCTIME	Synchronize date and time	***NONE**, *YES, *NO	Optional
PRPDMNUSR	Propagate domain user	***YES**, *NO	Optional
DSBUSRPRF	Disable user profile	***AUTO**, *NO	Optional
WNTLICKEY	Windows license key	*Character value*	Optional
LICMODE	License mode	*Element list*	Optional
	Element 1: License mode	***PERSEAT**, *PERUSER, *PERSERVER	
	Element 2: Client licenses	5-9999, ***NONE**	
	Element 3: Terminal services	***NONE**, *TSENABLE, *PERDEVICE, *PERUSER	
RSTDDEVRSC	Restricted device resources	Single values: ***NONE**, *ALL Other values (up to 10 repetitions): *Name*, *ALLTAPE, *ALLOPT	Optional
SHUTDTIMO	Shutdown timeout	2-45, **15**	Optional
ACTTMR	Activation timer	30-1800, **120**	Optional
CMNMSGQ	Communications message queue	Single values: ***SYSOPR** Other values: *Qualified object name*	Optional
	Qualifier 1: Communications message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
STGPTH	Storage path	*Element list*	Optional
	Element 1: Network server host adapter	*Name*	

Keyword	Description	Choices	Notes
POOL	Pool identifier	***BASE**, *SHRPOOL1, *SHRPOOL2, *SHRPOOL3, *SHRPOOL4, *SHRPOOL5, *SHRPOOL6, *SHRPOOL7, *SHRPOOL8, *SHRPOOL9, *SHRPOOL10, *SHRPOOL11, *SHRPOOL12, *SHRPOOL13, *SHRPOOL14, *SHRPOOL15, *SHRPOOL16, *SHRPOOL17, *SHRPOOL18, *SHRPOOL19, *SHRPOOL20, *SHRPOOL21, *SHRPOOL22, *SHRPOOL23, *SHRPOOL24, *SHRPOOL25, *SHRPOOL26, *SHRPOOL27, *SHRPOOL28, *SHRPOOL29, *SHRPOOL30, *SHRPOOL31, *SHRPOOL32, *SHRPOOL33, *SHRPOOL34, *SHRPOOL35, *SHRPOOL36, *SHRPOOL37, *SHRPOOL38, *SHRPOOL39, *SHRPOOL40, *SHRPOOL41, *SHRPOOL42, *SHRPOOL43, *SHRPOOL44, *SHRPOOL45, *SHRPOOL46, *SHRPOOL47, *SHRPOOL48, *SHRPOOL49, *SHRPOOL50, *SHRPOOL51, *SHRPOOL52, *SHRPOOL53, *SHRPOOL54, *SHRPOOL55, *SHRPOOL56, *SHRPOOL57, *SHRPOOL58, *SHRPOOL59, *SHRPOOL60	Optional
VRTETHPTH	Virtual Ethernet path	Values (up to 5 repetitions): *Element list*	Optional
	Element 1: Port	***VRTETHPTP**, *VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Network server host adapter	*Name*	
VRTETHCTLP	Virtual Ethernet control port	1024-65535, **8800**	Optional
RMTNWSCFG	Remote system NWSCFG	*Name*, ***DFT**	Optional
SPNWSCFG	Service processor NWSCFG	*Name*, ***DFT**	Optional
CNNNWSCFG	Connection security NWSCFG	*Name*, ***DFT**	Optional
INZSP	Initialize service processor	***NONE**, *SYNC	Optional
ENBUNICAST	Enable unicast	***YES**, *NO	Optional
EID	Enclosure identifier	Single values: ***AUTO** Other values: *Element list*	Optional
	Element 1: Serial number	*Character value*	
	Element 2: Manufacturer type and model	*Character value*	
SPNAME	Service processor name	*Character value*, ***SPINTNETA**	Optional
SPINTNETA	SP internet address	*Character value*	Optional
SPAUT	SP authentication	Single values: *DFT Other values: *Element list*	Optional
	Element 1: User name	*Character value*	
	Element 2: User password	*Character value*	
RMTSYSID	Remote system identifier	Single values: ***EID** Other values: *Element list*	Optional
	Element 1: Serial number	*Character value*	
	Element 2: Manufacturer type and model	*Character value*	
DELIVERY	Delivery method	***DYNAMIC**, *MANUAL	Optional
CHAPAUT	Target CHAP authentication	Single values: *NONE Other values: *Element list*	Optional
	Element 1: CHAP name	*Character value*, ***NWSCFG**	
	Element 2: CHAP secret	*Character value*, ***GEN**	

Keyword	Description	Choices	Notes
INRCHAPAUT	Initiator CHAP authentication	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: CHAP name	*Character value*, ***NWSCFG**	
	Element 2: CHAP secret	*Character value*, ***GEN**	
BOOTDEVID	Boot device ID	Single values: ***SINGLE** Other values: *Element list*	Optional
	Element 1: Bus	0-255	
	Element 2: Device	0-31	
	Element 3: Function	0-7	
DYNBOOTOPT	Dynamic boot options	*Element list*	Optional
	Element 1: Vendor ID	*Character value*, ***DFT**	
	Element 2: Alternate client ID	*Character value*, ***ADPT**	
RMTIFC	Remote (initiator) interfaces	*Element list*	Optional
	Element 1: SCSI interface	*Element list*	
	Element 1: Adapter address	*Hexadecimal value*	
	Element 2: Internet address	*Character value*	
	Element 3: Subnet mask	*Character value*	
	Element 4: Gateway address	*Character value*, ***NONE**	
	Element 5: iSCSI qualified name	*Character value*, ***GEN**	
	Element 2: LAN interface	*Element list*	
	Element 1: Adapter address	*Hexadecimal value*	
	Element 2: Internet address	*Character value*	
	Element 3: Subnet mask	*Character value*	
	Element 4: Gateway address	*Character value*, ***NONE**	
TEXT	Text ’description’	*Character value*, ***BLANK**	Optional
KBDTYPE	Keyboard layout	*Hexadecimal value*, ***DEFAULT**	Optional
CFGFILE	Configuration file	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Configuration file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CLU	Cluster name	*Name*, ***NONE**	Optional

Keyword	Description	Choices	Notes
CLUCFG	Cluster configuration	Single values: ***CLU** Other values: *Element list*	Optional
	Element 1: Cluster domain name	*Character value*	
	Element 2: Quorum resource size	550-1024000, ***CALC**	
	Element 3: Quorum resource ASP	1-255, **1**	
	Element 4: Quorum ASP device	*Name*	
	Element 5: Connection port	*VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 6: Cluster internet address	*Character value*	
	Element 7: Cluster subnet mask	*Character value*	
VRTPTPPORT	Virtual PTP Ethernet port	*Element list*	Optional
	Element 1: Internet address	*Character value*, ***GEN**	
	Element 2: Windows internet address	*Character value*, ***GEN**	

Top

Network server description (NWSD)

Specifies the name of the network server to be installed. The network server description is created using the values specified in this command. The Create Network Server Description (CRTNWSD) command will be used to create the NWSD. The name should also used as the computer name of the integrated server that is installed as well as the TCP host name for the integrated server.

This is a required parameter.

communications-name
Specify the name of the network server description.

The network server name can be up to eight characters. The following characters are allowed in NWSD names:

- Alphabetical characters A through Z
- Digits 0 through 9

Top

Installation type (INSTYPE)

Specifies the type of install to perform.

This is a required parameter.

***FULL** A full installation of the integrated server will be controlled by the system.

Note: Any integrated server that will be upgrading, OPTION(*UPGRADE), to a new version of Windows must also specify *FULL.

***BASIC**
A basic installation of the integrated server will be initiated by the system and completed using the ServerGuide CD.

Top

Resource name (RSRCNAME)

Specifies the resource name that identifies the hardware that the description uses.

This is a required parameter.

***ISCSI**
This network server resource is an iSCSI attached server using a network server host adapter device description. Use the Work with Device Descriptions (WRKDEVD) with *NWSH specified for the DEVD parameter to help determine which Network Server Host Adapters are configured.

name Specify the resource name of the File Server IOA communications adapter to use. Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

Top

Windows server version (WNTVER)

Specifies the integrated server operating system version to install on the integrated server. Use F4 while prompting the command to see the complete list of allowed values for this parameter.

This is a required parameter.

Top

Windows source directory (WNTSRCDIR)

Specifies the directory of the image that is used as the source for the install. The directory name may reference an optical volume (′/QOPT/volume′), or an IFS directory (′/dir1/dir2′).

To find out the name of a volume on an optical device, use the command: DSPOPT VOL(*MOUNTED) DEV(device-name). If you do not know the name of the optical device, use the command: WRKCFGSTS CFGTYPE(*DEV) CFGD(*OPT)

To find out the name of a path in an IFS directory, use the Work with Object Links (WRKLNK) command. WRKLNK will show the directory object path names on the system.

Use F4 while prompting the command to see a list of path names for optical volumes that are allowed values for this parameter. Folder or IFS path names that are not optical volumes are also allowed, but are not listed when using F4.

Note: Using *DFT or prompting with F4 on a system with an optical media library is not recommended. Every volume located in the ′/QOPT′ path will be searched for a valid server Operating System install source and will cause each volume to be retrieved and mounted. A specific optical volume path should be specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

Note: The specified directory must contain an I386 directory with a WINNT.EXE file in it.

***DFT** The default directory path name to use is determined by searching the QOPT file system (/QOPT directory in IFS).

path-name
Specify the server Operating System source path name to use for the install.

Top

Install option (OPTION)

Specifies the Windows server installation method. More information on each of type of install is available in the Windows server documentation. This information should be consulted prior to upgrading Windows server.

***INSTALL**
Install a new Windows server and the Integrated Server Support code.

This will create a new network server description, storage spaces, message queue, line descriptions, and TCP interfaces.

***UPGRADE**
Upgrade an existing Windows server and the Integrated Server Support code for a later release of Windows server.

This will use an existing network server description, system storage space, message queue, line descriptions, and TCP interfaces.

Notes:

1. A backup of all drives linked to the network server is strongly recommended before an *UPGRADE install is performed.
2. The install source drive (typically the D: drive) is deleted and recreated with the install source size specified on the **Server storage space sizes (SVRSTGSIZE)** parameter, **Install source size** element. Any user data on this drive will be lost!
3. An *UPGRADE install is the only supported upgrade path to a new release of Windows server. Upgrading the Windows server directly from an installation CD-ROM may cause the integrated server to become unusable and require that it be restored from a backup.

Top

TCP/IP port configuration (TCPPORTCFG)

Specifies the TCP/IP configuration values that are specific to a port on the integrated server.

This information consists of four parts including the identification of the integrated server port, the internet address, the subnet mask and the default gateway assigned to the port.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

Single values

***NONE**
There is no TCP/IP port configuration.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the locally managed integrated server port number.

1 Integrated server port number 1 is configured.

2 Integrated server port number 2 is configured.

3 Integrated server port number 3 is configured.

4 Integrated server port number 4 is configured.

Element 2: Windows internet address

character-value
Specify the internet address associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

An internet address that has a binary value of all ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the internet address is not valid.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 3: Windows subnet mask

character-value
Specify the subnet mask associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Windows gateway address

***NONE**
There is no gateway address.

character-value
Specify the gateway address associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Virtual Ethernet port (VRTETHPORT)

Specifies the TCP/IP configuration for the virtual Ethernet used by the integrated server.

Single values

***NONE**
Do not configure any virtual Ethernet ports.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the virtual Ethernet port number.

***VRTETHn**
The network server virtual Ethernet port 'n' is configured, where 'n' can have a value of 0 through 9.

Note: Each value can only be specified once.

Element 2: Windows internet address

character-value
Specify the internet address associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

An internet address that has a binary value of all ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the internet address is not valid.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 3: Windows subnet mask

character-value
Specify the subnet mask associated with the integrated server port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Associated port

Specifies the resource name that describes the port that is used to establish a connection between the integrated server and the network.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name. The resource name is on the port. For example, the resource name may be CMN01 on a Ethernet port.

***NONE**
An associated port resource name is not associated with the line.

name Specify the associated port resource name.

Top

TCP/IP local domain name (TCPDMNNAME)

Specifies the local domain name associated with the integrated server.

A domain name can be a text string having 1 to 255 characters. Domain names consist of one or more labels separated by periods. Each label can contain up to 63 characters. The following characters are allowed in domain names:

- Alphabetical characters A through Z
- Digits 0 through 9
- Minus sign (-)
- Underscore (_)

- Period (.). Periods are only allowed when they separate labels of domain style name (refer to RFC 1034).

Uppercase and lowercase characters are allowed, but no significance attached to the case. The case is maintained as entered. The first and last character of the host name must be an alphabetic character or a digit.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

***SYS** Specifies that the local domain name for the integrated server should be the same value as is configured for i5/OS.

character-value
Specify a TCP domain name to be associated with the integrated server.

Top

TCP/IP name server system (TCPNAMSVR)

Specifies the internet address of the name server system that is used by the integrated server. Typically, this is the same value as it is for i5/OS.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

Single values

***SYS** The name server systems used by the integrated server should be the same as for i5/OS.

***NONE**
There is no name server to be used by the integrated server.

Other values (up to 3 repetitions)

character-value
Specify an internet address for the name server system to be used by the integrated server. Up to three remote name server systems can be specified. The name server systems are used in the order they are specified.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Server message queue (MSGQ)

Specifies the message queue to receive integrated server messages.

Integrated Server Support messages and event logs received from the server are logged.

This message queue should be monitored so that it does not become full. If it becomes full, messages will be rerouted to the joblog of the monitor job.

Care should be taken if QSYSOPR is specified as the message queue to receive all messages related to Windows server running on an integrated server because the volume of Windows server event log messages is unpredictable.

If both a message queue name and library name is specified, and the message queue does not exist, the message queue will automatically be created with authority *EXCLUDE. If the library specified for the message queue does not exist, the command will fail.

Single values

*JOBLOG
Messages are placed on the joblog of the monitor job.

***NONE**
Messages are not placed on any message queue.

Qualifier 1: Server message queue

name Specify the name of the message queue to receive messages issued by the integrated server.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread 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.

name Specify the name of the library to be searched.

Top

Event log (EVTLOG)

Specifies whether or not messages from the event logs are received from the server.

Note: Event log messages are placed in the message queue that is specified for the **Server message queue (MSGQ)** parameter. The MSGQ value cannot be *NONE if a value other than *NONE is specified for this parameter. See the MSGQ parameter description for more information.

Single values

*ALL All the event log messages are received.

***NONE**
No event log messages are received.

Other values (up to 3 repetitions)

Note: Each value can only be specified once.

***SYS** The system event log messages are received.

***SEC** The security event log messages are received.

***APP** The application event log messages are received.

Top

Server storage space sizes (SVRSTGSIZE)

Specifies the size of the server storage spaces, in megabytes.

Notes:

1. The install source drive contains the contents of the Integrated Server Support code. The contents of the I386 directory of the Windows server installation media is also copied to the install source drive for Windows 2000 and Windows Server 2003. The size specified for the install source drive must be large enough to hold this data.
2. For an OPTION(*UPGRADE) install, a new install source drive size may be specified. The new drive that is created will replace the existing install source drive (typically the D: drive) resulting in the loss of all user data on this drive. A backup of all drives associated with this network server is recommended.
3. The minimum size allowed is determined by the Windows server version specified on the WNTVER parameter.
4. A value other than the default for the System size may not be specified when OPTION is *UPGRADE.

Element 1: Install source size

Specifies the size of the storage space that holds the files used to install the integrated server operating system.

***CALC**
Specify that the system should calculate the size based on the space required to install the integrated server.

200-2047
Specify the install source size value in megabytes.

Element 2: System size

Specifies the size in megabytes of the storage space that the integrated server operating system is installed on.

***CALC**
Specify that the size should be calculated by the system based on the values specified on other parameters.

1024-1024000
Specify the system size value in megabytes.

Top

Storage space ASP (SVRSTGASP)

Specifies the auxiliary storage pool (ASP) identifiers for the storage space that will contain the files used to install the integrated server and the storage space that will contain the integrated server operating system.

Note: You cannot specify both a SVRSTGASP and STGASPDEV parameter value for the same element.

Note: Null (omitted) values are specified with the characters *N, which mean that no value was specified. The value specified for the corresponding Server storage ASP device (STGASPDEV) element will be used if specified. Otherwise, the default value will be used. *N is needed only when another value following the omitted element is being specified.

Note: A value other than the default for the System size may not be specified when OPTION is *UPGRADE.

Element 1: Install source ASP

Specifies the auxiliary storage pool for the storage space that holds the files that are used to install the integrated server operating system.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

2-255 Specify the number of the ASP to be used. Valid values depend on what ASPs are defined on the system.

Element 2: System ASP

Specifies the auxiliary storage pool for the storage space that holds the integrated server operating system.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

2-255 Specify the number of the ASP to be used. Valid values depend on what ASPs are defined on the system.

Top

Server storage ASP device (STGASPDEV)

Specifies the auxiliary storage pool (ASP) device name for the storage space that will contain the files used to install the integrated server operating system and the storage space that will contain the integrated server operating system.

Note: You cannot specify both a SVRSTGASP and STGASPDEV parameter value for the same element.

Note: The ASP must have been activated (by varying on the ASP device) and have a status of ′Available′.

Element 1: Install source ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the files used to install the integrated server operating system.

name The device name of the ASP to use for the network server storage space.

Element 2: System ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the integrated server operating system.

name The device name of the ASP to use for the network server storage space.

Top

Convert to NTFS (CVTNTFS)

Specifies if the primary partition of the system drive should be converted to the NT File System (NTFS).

Note: Conversion to NTFS may be automatically performed under some situations such as the File system limitations for system drives greater than 32000 megabytes. When one of these conditions exist, the Convert to NTFS (CVTNTFS) parameter is automatically set to *YES by this command.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

***YES** Windows server will convert the system drive to NTFS during the install.

***NO** Windows server will not convert the system drive to NTFS during the install.

Top

To workgroup (TOWRKGRP)

Specifies the workgroup in which the computer will participate. The Windows server will prompt for a value during the install if no parameter value is specified.

Characters allowed for this parameter include any upper-case character A-Z, lower-case a-z, 0-9, and any character in the ASCII code page 850 except:

- space ( )
- quote (″)
- asterisk (*)
- plus (+)
- comma (,)
- period (.)
- forward slash (/)
- colon (:)
- semicolon (;)
- less-than (<)
- equal (=)
- greater-than (>)
- question mark (?)
- open square bracket ([)
- backward slash (\)
- close square bracket (])
- vertical bar (|)

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

character-value
Specify the workgroup in which the computer will participate. The computer can be part of either a workgroup or a domain.

Top

To domain (TODMN)

Specifies the name of an existing server domain in which the computer will participate. Windows server will prompt for a value during the install if no parameter value is specified.

Characters allowed for this parameter include any upper-case character A-Z, lower-case a-z, 0-9, and any character in the ASCII code page 850 except:

- space ( )
- quote (″)
- asterisk (*)
- plus (+)
- comma (,)
- period (.)
- forward slash (/)
- colon (:)
- semicolon (;)
- less-than (<)
- equal (=)
- greater-than (>)
- question mark (?)
- open square bracket ([)
- backward slash (\)
- close square bracket (])
- vertical bar (|)

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

character-value
Specify the name of the server domain in which the computer will participate. The computer can be part of either a workgroup or a domain.

Top

Full Name (FULNAM)

Specifies the users' full name for the Windows server being installed. Windows server will prompt for a value during the install if no parameter value is specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

character-value
Specify the name of the user installing Windows server who holds the license.

Top

Organization (ORG)

Specifies the organization name for the Windows server being installed. Windows server will prompt for a value during the install if no parameter value is specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

character-value
Specify the name of the organization installing Windows server who holds the license.

Top

Language version (LNGVER)

Specifies the installed language environment used to display Integrated Server Support text and messages. Note that a smaller set of languages are available for messages displayed during installation than for messages displayed after installation.

***PRIMARY**
The installed language environment for Integrated Server Support text and messages is based on the language feature of the system primary language.

integer
Specify the language feature that will be used to select the Integrated Server Support text and messages.

Top

Synchronize date and time (SYNCTIME)

Specifies whether i5/OS should synchronize the integrated server date and time with i5/OS date and time.

The QTIMZON system value must be set to the correct value for time synchronization to work correctly.

***NONE**
i5/OS will never synchronize the integrated server date and time with the i5/OS date and time when the network server description is varied on.

***YES** i5/OS will synchronize integrated server date and time with i5/OS date and time.

***NO** i5/OS only synchronizes the integrated server date and time with i5/OS date and time once when the network server description is varied on.

Top

Propagate domain user (PRPDMNUSR)

Specifies if this server should be used to propagate and synchronize users to the Windows domain or active directory.

Note: When multiple network servers belong to the same Windows domain, only one needs to propagate users to the domain. Selecting a network server with a domain role of *DMNCTL will provide the fastest performance and may eliminate the need for the special QAS400NT userid. At least one network server should specify *YES for each Windows domain that you wish to propagate users to.

***YES** Propagation of domain user enrollment is allowed for this network server description.

***NO** Propagation of domain user enrollment is not allowed for this network server description.

Top

Disable user profile (DSBUSRPRF)

Specifies whether to disable the integrated servers user profiles if the corresponding i5/OS user profiles are disabled.

***AUTO**
Integrated server user profiles are disabled if corresponding i5/OS user profiles are disabled.

***NO** Integrated server user profiles are not disabled if corresponding i5/OS user profiles are disabled.

Top

Windows license key (WNTLICKEY)

Specifies the license key for Windows server. Windows server will prompt for a value during the install if no parameter value is specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

character-value
Specify the license key for the Windows server, including any dashes ('-'). The length of the Windows license key (WNTLICKEY) is limited to 34 characters.

Top

License mode (LICMODE)

Determines the license mode to install the Windows server as.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

Element 1: License mode

***PERSEAT**
Indicates that the end user purchased a client access license for each device or user accessing the server.

***PERUSER**
Indicates that the end user purchased a client access license for each device or user accessing the Windows Server 2003 server.

***PERSERVER**
Indicates that the end user purchased client access licenses for each server, which allows a certain number of concurrent connections to the server.

Element 2: Client licenses

***NONE**
Indicates that no client licenses are installed. *NONE must be specified when *PERSEAT is specified.

integer
Number of client licenses purchased for the server being installed. A number greater than five must be specified when *PERSERVER is specified for the license type. The valid range is from 5 to 9999.

Element 3: Terminal services

***NONE**
Do not install the Terminal Server component for this server.

***TSENABLE**
Installs Terminal Services on Windows 2000.

***PERDEVICE**
Installs and configures the Windows Server 2003 Terminal Services to require that each connected device has a valid Terminal Server Client Access License (CAL). If the client has a Terminal Server CAL, it can access more than one Terminal Server.

***PERUSER**
Installs and configures Windows Server 2003 Terminal Server to provide one Terminal Server CAL for each active user.

Top

Restricted device resources (RSTDDEVRSC)

Specifies the tape and optical device resource names that are restricted and cannot be used by the integrated server.

The resource is used when the integrated server is active and a request is issued from a client that is running an application. The device resource cannot be used by the client application and i5/OS based application at the same time. If the device resource is intended to be used by an application, it will need to be available when the application is ready to use it.

Single values

***NONE**
No device resources are restricted from the integrated server. Therefore, any tape or optical device resources that exist on the system can be used.

***ALL** All tape and optical device resources are restricted from being used by the integrated server.

Other values (up to 10 repetitions)

***ALLOPT**
All optical device resources are restricted from being used by the integrated server.

Note: This value can only be specified once.

***ALLTAPE**
All tape resources are restricted from being used by the integrated server.

Note: This value can only be specified once.

name Specify the resource name of a restricted device that cannot be used by the integrated server. Up to 10 restricted device resource names can be specified.

Top

Shutdown timeout (SHUTDTIMO)

Specifies the shutdown timeout value in minutes. This is used to limit the amount of time that the integrated server operating system is allowed to shutdown before the network server description is varied off.

15 The integrated server default shutdown time-out value is used.

2-45 Specify the number of minutes to wait. i5/OS waits until the integrated server operating system has shutdown successfully, or until the specified time passes before varying the network server description off.

Top

Activation timer (ACTTMR)

Specifies the amount of time (in seconds) the system will wait for the connection to be established to the remote server service processor and to power on the remote server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

120 The activate time of 120 seconds is used.

integer
Specify, in seconds, a value ranging from 30 through 1800.

Top

Communications message queue (CMNMSGQ)

Specifies the name of a message queue to receive network server host adapter communications status messages.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Single values

***SYSOPR**
Causes network server host adapter communications status messages to be placed in the system operator message queue.

Qualifier 1: Communications message queue

name Specify the name of a message queue to receive network server host adapter communications status messages.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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.

name Specify the name of the library to be searched.

Top

Storage path (STGPTH)

Specifies the storage path the storage spaces can use.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Element 1: Network server host adapter

name Specify the name of an existing **Network server host adapter** (NWSH) device description.

Top

Pool identifier (POOL)

Specifies the shared data storage pool this integrated server should use.

***BASE**
The base pool is to be used by this integrated server.

***SHRPOOLnn**
Specifies the shared pool to be used by this integrated server. There are sixty general-purpose shared pools, identified by special values *SHRPOOL1 to *SHRPOOL60.

Top

Virtual Ethernet path (VRTETHPTH)

Specifies the virtual Ethernet paths the Ethernet line descriptions can use. This information consists of two parts including the virtual Ethernet port and the **Network server host adapter** (NWSH) description. You must enter at least one virtual Ethernet path which is the path to be used by the *VRTETHPTP line description.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

You can specify 5 values for this parameter.

Element 1: Port

***VRTETHPTP**
The network server virtual Ethernet point to point port is configured.

***VRTETHn**
The virtual Ethernet port ’n’ is configured, where ’n’ can have a value of 0 through 9.

Element 2: Network server host adapter

name Specify the name of an existing **Network server host adapter** (NWSH) description. The network server host adapter name does not need to be unique for each VRTETHPTH parameter on this NWSD.

Top

Virtual Ethernet control port (VRTETHCTLP)

Specifies the TCP port to use for virtual Ethernet control.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

8800 Use the TCP port number of 8800.

1024-65535
Specify the port number identifying the port that is to be used for virtual Ethernet control.

Top

Remote system NWSCFG (RMTNWSCFG)

Specifies the remote system network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default remote system network server configuration name of ′nwsdnameRM′ where nwsdname is the name of the network server description.

name Specify the name of an existing remote system network server configuration.

Top

Service processor NWSCFG (SPNWSCFG)

Specifies the service processor network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default service processor network server configuration name of ′nwsdnameSP′ where nwsdname is the name of the network server description.

name Specify the name of an existing service processor network server configuration.

Top

Connection security NWSCFG (CNNNWSCFG)

Specifies the connection security network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default connection security network server configuration name of ′nwsdnameCN′ where nwsdname is the name of the network server description.

name Specify the name of an existing connection security network server configuration.

Top

Initialize service processor (INZSP)

Specifies how the remote system service processor is secured.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***NONE**

Provides no security.

Use this only if the interconnecting network is physically secure.

Note: Some service processors do not support secure connections. Use *NONE for these service processors. Additional information can be found at System i integration with BladeCenter and System x at http://www.ibm.com/systems/i/bladecenter/.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***SYNC**

Synchronize the user name and password and self-signed certificate from the service processor. This option is used to initially synchronize i5/OS with the service processor. It is also used if multiple service processor network server configurations are used for the same system or the service processor network server configuration has been restored from backup and the service processor certificate must be synchronized. The current user name and password for the service processor must be specified on the **SP authentication (SPAUT)** parameter to perform this option.

Top

Enable unicast (ENBUNICAST)

Specifies whether unicast packet distribution is to be used. Unicast is a transmission method where packets are sent directly to the specified **Service processor name (SPNAME)** or **SP internet address (SPINTNETA)** parameter.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***YES** Enable unicast.

***NO** Disable unicast

Top

Enclosure identifier (EID)

Specifies the identifying serial number, type and model of the enclosure containing the service processor.

When specified, they are used to locate the system on the network.

Look for these values on the label of the system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

***AUTO**
Automatically retrieve the identifier when ENBUNICAST(*YES) is specified.

Element 1: Serial number

character-value
Specify the machine serial number.

Element 2: Manufacturer type and model

character-value
Specify the machine type and model.

The value is entered in the form ttttmmm where tttt is the machine type and mmm is the machine model number.

Top

Service processor name (SPNAME)

Specifies the remote system service processor host name.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Notes:

1. This parameter is required when ENBUNICAST(*YES) is specified.
2. This parameter is ignored when ENBUNICAST(*NO) is specified.

***SPINTNETA**

The remote system is identified by the value specified for the **SP internet address (SPINTNETA)** parameter.

character-value
Specify the remote system service processor host name.

Top

SP internet address (SPINTNETA)

Specifies the remote system service processor internet address.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Notes:

1. This parameter is ignored when ENBUNICAST(*NO) is specified.
2. This parameter is required when SPNAME(*SPINTNETA) is specified.

character-value
Specify the internet address of the service processor.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

SP authentication (SPAUT)

Specifies the service processor user name and password. This is used to authenticate and secure the service processor.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

***DFT** The default service processor userid and password are used.

Element 1: User name

character-value
Specify a name that represents the host configuration that owns the service processor. It is suggested that the remote system network server configuration name be used. If multiple remote system network server configurations use the same service processor at different times, each configuration must contain the same user name and password.

Element 2: User password

character-value
Specify the service processor password. Password must be at least 5 characters in length and contain at least one alphabetic character and one numeric or symbolic character.

Top

Remote system identifier (RMTSYSID)

Specifies the identifying serial number, type and model of the remote system. When specified, they are used to locate the remote system on the network.

Look for these values on the label of the system.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Single values

***EID** Use the service processor identifier.

Element 1: Serial number

character-value
Specify the machine serial number.

Element 2: Manufacturer type and model

character-value
Specify the machine type and model.

The value is entered in the form ttttmmm where tttt is the machine type and mmm is the machine model number.

Top

Delivery method (DELIVERY)

Specifies how the parameters necessary to configure the remote system are delivered.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

***DYNAMIC**
Parameters are dynamically delivered to the remote system using Dynamic Host Configuration Protocol (DHCP).

***MANUAL**
Parameters are manually configured on the remote system using the BIOS utilities (System BIOS or Adapter BIOS - CTRL-Q).

Top

Target CHAP authentication (CHAPAUT)

Specifies the Challenge Handshake Authentication Protocol (CHAP) for the System i iSCSI target to authenticate the remote system iSCSI initiators.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***NONE**
CHAP authentication is not enabled.

Element 1: CHAP name

***NWSCFG**
The system will automatically generate a name for CHAP using the Network server configuration name.

character-value
Specify the name you want to use for the Challenge Handshake Authentication Protocol.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Element 2: CHAP secret

***GEN** The system will automatically generate a random CHAP secret.

character-value

Specify the secret you want to use for the Challenge Handshake Authentication Protocol.

Note: Target and initiator CHAP secrets must not be the same.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Top

Initiator CHAP authentication (INRCHAPAUT)

Specifies the Challenge Handshake Authentication Protocol (CHAP) for the remote system iSCSI initiators to authenticate the System i iSCSI target.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***NONE**
CHAP authentication is not enabled.

Element 1: CHAP name

***NWSCFG**
The system will automatically generate a name for CHAP using the Network server configuration name.

character-value
Specify the name you want to use for the Challenge Handshake Authentication Protocol.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Element 2: CHAP secret

***GEN** The system will automatically generate a random CHAP secret.

character-value
Specify the secret you want to use for the Challenge Handshake Authentication Protocol.

Note: Target and initiator CHAP secrets must not be the same.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Top

Boot device ID (BOOTDEVID)

Specifies the PCI Function Address (Bus/Device/Function) of the iSCSI initiator port in the remote system that will be used to boot from.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***SINGLE**
The single iSCSI initiator port is used on the remote system

Element 1: Bus

0-255 Specify the bus number of the remote system iSCSI initiator port that will be used to boot.

Element 2: Device

0-31 Specify the device number of the remote system iSCSI initiator port that will be used to boot.

Element 3: Function

0-7 Specify the function number of the remote system iSCSI initiator port that will be used to boot.

Top

Dynamic boot options (DYNBOOTOPT)

Specifies the internal Dynamic Host Configuration Protocol (DHCP) Server configuration.

Note: This is an advanced configuration function.

This parameter is used to configure the internal DHCP Server that is part of the iSCSI target host bus adapter (HBA) firmware. It is used to provide IP address and diskless boot parameters for the remote system iSCSI initiator port.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

This parameter is only valid when DELIVERY(*DYNAMIC) is specified.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Element 1: Vendor ID

The client and server are pre-configured to a default vendor ID. Network administrators can configure clients to define their own identifying values to convey hardware, operating system or other identifying information. DHCP option 60 described in the IETF RFC 2132 is used for this function.

***DFT** The default vendor ID will be used.

character-value
Vendor ID of the remote system iSCSI initiator port that will be used.

Element 2: Alternate client ID

Used by clients to specify their unique identifier to the server. Each client identifier must be unique among all other client identifiers used on the effective DHCP network to which the client is attached (that is, the client local subnet and any remote subnets reachable using DHCP relay). Vendors and system administrators are responsible for choosing client identifiers that meet this requirement for uniqueness. DHCP option 61 described in the IETF RFC 2132 is used for this function.

***ADPT**
The default Client ID consists of the adapter address for the remote system iSCSI initiator port. This value will be used to identify the remote system.

character-value
Specify the Client ID of the remote system iSCSI initiator port that will be used to boot.

Top

Remote (initiator) interfaces (RMTIFC)

Specifies the remote system iSCSI initiator host bus adapter (HBA) configuration. Each iSCSI initiator port has two functions to support a SCSI and a LAN or TCP Offload Engine (TOE) interface.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Element 1: SCSI interface

Specifies the SCSI interface.

Element 1: Adapter address

hexadecimal-value
Specify the 12-character hexadecimal adapter address for the initiator port.

Element 2: Internet address

character-value
Specify the internet address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

character-value
Specify the subnet mask for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

***NONE**
No gateway address is configured for this initiator port.

character-value
Specify the gateway address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 5: iSCSI qualified name

***GEN** The system will automatically generate the iSCSI qualified name.

character-value
Specify the iSCSI qualified name for the initiator port.

The following characters are allowed in iSCSI qualified names:

- Alphabetical characters A through Z converted to lower case (refer to RFC 3722)
- Alphabetical characters a through z
- Digits 0 through 9
- Period (.)
- Dash (-)
- colon (:)

Element 2: LAN interface

Specifies the LAN interface.

Element 1: Adapter address

hexadecimal-value
Specify the 12-character hexadecimal adapter address for the initiator port.

Element 2: Internet address

character-value
Specify the internet address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

character-value
Specify the subnet mask for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

***NONE**
No gateway address is configured for this initiator port.

character-value
Specify the gateway address for the initiator port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Text ’description’ (TEXT)

Specifies text that briefly describes the network server configuration.

***BLANK**
Text is not specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Keyboard layout (KBDTYPE)

Specifies the keyboard layout identifier to install on the Windows server. The valid keyboard layout identifiers are listed in the TXTSETUP.SIF file in the I386 directory of the Windows server installation media.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

***DEFAULT**
The default keyboard layout for the version of Windows server being installed is used.

hexadecimal-value
Specify the keyboard layout identifier to be used by the Windows server.

Top

Configuration file (CFGFILE)

Specifies the source file containing configuration data to be used in activating or further defining the integrated server.

Single values

***NONE**
No configuration file is specified.

Qualifier 1: Configuration file

name Specify the name of the source file containing the configuration data members for the server. At the time the server is activated, all members in the file are processed. The file must exist on the system by the time the server is activated.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread 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.

name Specify the name of the library to be searched.

Top

Cluster name (CLU)

Specifies the name of the cluster.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

***NONE**
Do not form or join a Windows Cluster.

name Specify the name of the cluster. Administrators will use this name for connections to the cluster. The cluster name must be different from the domain name, from all computer names on the domain, and from other cluster names on the domain.

Top

Cluster configuration (CLUCFG)

Specifies the parameters required to configure a new Windows Cluster.

Note: This parameter is only required when forming a new Windows cluster using the **Cluster name (CLU)** parameter.

Note: This parameter is only allowed when WNTSVR is *WIN2000 or *WIN2003.

Single values

***CLU** Use the values already defined by a previous installation of the Windows Cluster service to join the existing cluster name.

Element 1: Cluster domain name

Specifies the domain to which the cluster belongs. If the cluster already exists, the cluster will be joined, otherwise, the cluster will be formed. If forming a cluster, the **Cluster configuration (CLUCFG)** parameter must be specified.

character-value
Specify the domain name to which the cluster belongs when forming a new cluster.

Element 2: Quorum resource size

Specifies the size in megabytes of the storage space used as the Windows quorum resource.

***CALC**
Specifies that the size should be calculated to be the default value based on the **Windows server version (WNTVER) parameter**.

550-1024000
Specify the Windows quorum resource size in megabytes. The size must be at least 550 megabytes but no larger than 1024000 megabytes.

Element 3: Quorum resource ASP

Specifies the auxiliary storage pool for the storage space used as the Windows quorum resource.

Note: You cannot specify both a Quorum resource ASP and a Quorum ASP device value.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

2-255 Specify the number of the ASP to be used. Valid values depend on what ASPs are defined on the system.

Element 4: Quorum ASP device

Specifies the independent auxiliary storage pool device name for the storage space used as the Windows quorum resource.

Note: You cannot specify both a Quorum resource ASP and a Quorum ASP device value.

name Specify the name of the independent auxiliary storage pool device. Valid values depend on which IASPs are defined on the system.

Element 5: Connection port

Specifies the connection port used for the Cluster service communication.

***VRTETHn**
The network server virtual Ethernet port 'n' is configured, where 'n' has a value of 0 through 9.

Element 6: Cluster internet address

Specifies the internet address for the cluster.

character-value
Specify the Cluster internet address.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 7: Cluster subnet mask

character-value
Specify the subnet mask for the Cluster internet address.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Virtual PTP Ethernet port (VRTPTPPORT)

Specifies the TCP/IP configuration for the virtual point-to-point Ethernet port.

Notes:

1. The VRTPTPPORT parameter must be used for integrated servers running on the host system.
2. The subnet mask that is used for both sides of the virtual point-to-point Ethernet port is 255.255.255.0 by default. Therefore, the internet addresses that are chosen for both sides of the virtual point-to-point Ethernet port must have the same values for the first three parts of the internet addresses.

Element 1: Internet address

Specifies the internet address for the i5/OS side of the virtual point-to-point Ethernet connection.

***GEN** Specify *GEN to let the system configure a virtual point-to-point Ethernet port with a generated internet address.

character-value
Specify the i5/OS internet address for the virtual point-to-point Ethernet port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Element 2: Windows internet address

Specifies the internet address for the integrated server side of the virtual point-to-point Ethernet connection.

***GEN** Specify *GEN to let the system configure a virtual point-to-point Ethernet port with a generated internet address.

character-value
Specify the integrated server internet address for the virtual point-to-point Ethernet port.

The value is specified in the form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all network server descriptions and the i5/OS TCP/IP configuration.

Top

Examples

Example 1: Installing a Windows Terminal Server

```
INSWNTSVR  NWSD(W2KSERV) INSTYPE(*FULL)
             RSRCNAME(LIN09) DMNROLE(*SERVER)
             WNTVER(*WIN2000) WNTSRCDIR(*DFT) OPTION(*INSTALL)
             TCPPORTCFG((1 '206.5.8.48' '255.255.255.128'
                         '206.5.8.1'))
             SVRSTGSIZE(*CALC 2500) SVRSTGASP(1 1)
             CVTNTFS(*YES) TOWRKGRP(XYZGROUP)
             FULNAM('John Smith') ORG('XYZ Corporation')
             WNTLICKEY('VVVVV-WWWWW-XXXXX-YYYYY-ZZZZZ')
             LICMODE(*PERSEAT *NONE *TSENABLE)
             TEXT('Windows 2000 Terminal Server')
```

This command installs a Windows server named W2KSERV. W2KSERV is the network server description associated with the Windows 2000 Server operating system that will be installed on the Integrated xSeries Server resource LIN09.

A fully controlled install will be performed with a system drive size of 2500 MB that will automatically be converted to NTFS during the install. The Windows 2000 Server will join the XYZGROUP workgroup. A per seat license mode will be configured and Terminal Services will be installed on the server.

The TCP/IP local host name is the same as the server description name. The TCP/IP local domain name is the same as the hosting system and the same name servers will be used. The TCP/IP address 206.5.8.48 will automatically be configured for the first LAN adapter detected on the Integrated xSeries Server.

Example 2: Installing a Windows Cluster Node

```
INSWNTSVR  NWSD(W2KNODE1) INSTYPE(*FULL)
            RSRCNAME(LIN03)
            WNTVER(*WIN2000) WNTSRCDIR(*DFT) OPTION(*INSTALL)
            TCPPORTCFG((1 '206.5.8.60' '255.255.255.128'
                        '206.5.8.1'))
            VRTETHPORT((*VRTETH5 '192.168.9.3'
                        '255.255.255.0'))
            TCPDMNNAME(xyzdomain.xyzcorp.com)
            TCPNAMSVR('206.5.69.165' '206.5.8.8'
                      '206.10.244.100')
            SVRSTGSIZE(*CALC 4000) SVRSTGASP(1 1)
            CVTNTFS(*YES) TODMN(XYZDOMAIN)
            FULNAM('John Smith') ORG('XYZ Corporation')
            WNTLICKEY('VVVVV-WWWWW-XXXXX-YYYYY-ZZZZZ')
            LICMODE(*PERSEAT *NONE *NONE)
            TEXT('Windows 2000 Cluster Node 1')
            CLU(CLU1XYZ)
            CLUCFG(XYZDOMAIN 600 1 *N *VRTETH5 '206.5.8.65'
                  '255.255.255.128')
```

This command installs a Windows server named W2KNODE1. W2KNODE1 is the network server description associated with the Windows 2000 Server operating system that will be installed on the Integrated xSeries Server resource LIN03.

A fully controlled install will be performed with a system drive size of 4000 MB that will automatically be converted to NTFS during the install. The Windows 2000 Server will join the XYZDOMAIN domain. A per seat license mode will be configured on the server. The TCP/IP address 206.5.8.60 will automatically be configured for the first LAN adapter detected on the Integrated xSeries Server. A virtual ethernet LAN will be created on virtual ethernet 5 configured with TCP/IP address 192.168.9.3.

The TCP/IP local host name is the same as the server description name. The TCP/IP local domain name is xyzdomain.xyzcorp.com. The TCP/IP name servers will use 206.5.69.165 206.5.8.8 and 206.10.244.100.

A new Microsoft Cluster will be enabled by creating a quorum resource drive named XYZDOMAIN that is 600 MB is size. Virtual ethernet 5 will be used for the private communication between clustered nodes.

Example 3: Installing Windows Server 2003 on an iSCSI attached server

```
INSWNTSVR  NWSD(WS03LAN) INSTYPE(*FULL)
            RSRCNAME(*ISCSI)
            WNTVER(*WIN2003) WNTSRCDIR(*DFT) OPTION(*INSTALL)
            TCPPORTCFG((1 '206.5.8.68' '255.255.255.128'
                        '206.5.8.1'))
            SVRSTGSIZE(*CALC 2500) SVRSTGASP(1 1)
            CVTNTFS(*YES) TOWRKGRP(XYZGROUP)
            FULNAM('John Smith') ORG('XYZ Corporation')
            WNTLICKEY('VVVVV-WWWWW-XXXXX-YYYYY-ZZZZZ')
            STGPTH(NWSHRG1)
            POOL(*SHRPOOL1)
            VRTETHPTH((*VRTETHPTP NWSHRG1))
            INZSP(*SYNC)
            ENBUNICAST(*NO)
            EID(1234567 418477U)
            SPAUT(spadmin spuid)
            RMTSYSID(*EID)
            DELIVERY(*DYNAMIC)
            CHAPAUT(mychapid mychapsecret)
            RMTIFC((020134304760 '206.5.8.92' '255.255.255.128'
                    '6.5.8.1')
                   (020134604750 '206.5.8.96' '255.255.255.128'
                    '6.5.8.1'))
            TEXT('Windows Server 2003 iSCSI Server')
```

This command installs a Windows server named WS03LAN. WS03LAN is an iSCSI attached server using Windows Server 2003. Network server host adapter (NWSH) device NWSHRG1 is configured for the storage and virtual Ethernet path.

Shared data pool *SHRPOOL1 will be used by the integrated server to handle virtual disk requests.

The service processor is automatically configured and is located using the enclosures serial number 1234567 and type/model 418477U.

The remote system is dynamically configured and secured using a specified CHAP name and secret.

Remote SCSI and LAN interfaces for the remote systems iSCSI initiator ports are configured.

Example 4: Installing Windows Server 2007 on an iSCSI attached server

```
INSWNTSVR  NWSD(WS07SRV) INSTYPE(*FULL)
           RSRCNAME(*ISCSI)
           WNTVER(*WIN2007) OPTION(*INSTALL)
           SVRSTGSIZE(*CALC 12500) SVRSTGASP(1 1)
           STGPTH(NWSHRG2)
           POOL(*SHRPOOL9)
           VRTETHPTH((*VRTETHPTP NWSHRG2))
           INZSP(*SYNC)
           SPNAME('bldchs1')
           EID(*AUTO)
           SPAUT(spadmin spuid)
           RMTSYSID(KQTHTYX 884325U)
           CHAPAUT(*NONE)
           DELIVERY(*DYNAMIC)
           RMTIFC((020134304760 '206.5.8.92' '255.255.255.128'
                  '6.5.8.1')
                 (020134604750 '206.5.8.96' '255.255.255.128'
                  '6.5.8.1'))
           TEXT('Windows Server 2007 iSCSI Server')
```

This command installs a Windows server named WS07SRV. WS07SRV is an iSCSI attached server using Windows Server 2007. Network server host adapter (NWSH) device NWSHRG2 is configured for the storage and virtual Ethernet path.

Shared data pool *SHRPOOL9 will be used by the integrated server to handle virtual disk requests.

The service processor is automatically configured and is located using the bldchs1 host name.

The remote system is identified by the serial number KQTHTYX and type/model 884325U.

Remote SCSI and LAN interfaces for the remote systems iSCSI initiator ports are configured.

Top

Error messages

ESCAPE Messages

NTA1007

Network server description &1 must be varied off.

NTA100E
Vary on or off of the Windows server not successful.

NTA1013
Integrated server installation not successful.

NTA1024
Storage space assigned to network server &1 missing, damaged or not valid.

NTA1030
Internal error occurred.

Top

Initialize DLFM (INZDLFM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Initialize DLFM (INZDLFM) command prepares the DataLink File Manager (DLFM) to be started, and clears information from the database files used by the DLFM.

Restrictions:

- To use this command, you must have input/output system configuration (*IOSYSCFG) special authority.

Top

Parameters

Keyword	Description	Choices	Notes
CLEARDB	Clear existing databases	***LNKSTS**, *ALL	Optional, Positional 1

Top

Clear existing databases (CLEARDB)

Specifies which databases should be cleared.

***LNKSTS**
The database files containing link status of DataLinks will be cleared. Database files containing registered prefixes and host database names will not be cleared.

***ALL** All database files used by the DataLink File Manager (DLFM) will be cleared.

Top

Examples

Initializing and Clearing a DataLink File Manager

```
INZDLFM   CLEARDB(*ALL)
```

This command initializes the DataLink File Manager, and clears all database files of existing data.

Top

Error messages

***ESCAPE Messages**

CPF3168

DataLink File Manager (DLFM) command failed.

Top

Initialize Distribution Queue (INZDSTQ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Initialize Distribution Queue (INZDSTQ) command resets the status of a distribution queue and the entries on the queue. It also optionally clears all distributions on the queue. This command applies to both the normal and high priority sections of the specified queue.

Attention: Initializing a distribution queue can result in the loss or duplication of distributions in the network, depending on the status of the distributions in transit at the time this command is run.

Initializing a distribution queue includes the following:

- If a SNADS (Systems Network Architecture (SNA) distribution services) sender job is active for the queue, the job is ended. This job cancelation takes effect immediately. Distribution queues being sent are interrupted.
- If the queue type is a SystemView distribution services (SVDS) queue type and a receiver job is active for this connection, the job is ended. This job cancelation takes effect immediately. All partially received distributions are discarded.
- If the distribution queue is to be cleared, all distributions on the queue are deleted as specified on the **Clear queue entries** prompt (CLEAR parameter).
- If the queue is not cleared, the distributions on the queue that do not have ″Held″ status are set to ″Ready.″ Distributions with a status of ″Held″ remain held.
- The queue status is set to ″Ready″ unless the queue is in the ″Held″ status.
- If the QSNADS system is active, a SNADS sender job is submitted for the queue following the same rules used to start the QSNADS subsystem.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	*Character value*	Required, Positional 1
CLEAR	Clear queue entries	***NO**, *YES, *PURGE	Optional

Top

Distribution queue (DSTQ)

Specifies the name of the distribution queue to initialize. The queue must be previously configured using the Configure Distribution Services (CFGDSTSRV) or the Add Distribution Queue (ADDDSTQ) command.

This is a required parameter.

Top

Clear queue entries (CLEAR)

Specifies whether distributions on the queue are deleted.

Attention: Using the *PURGE value results in the loss of distributions with no trace.

The possible values are:

<u>*NO</u> Distributions on the queue are not deleted.

***YES** Distributions on the queue are deleted. Each deleted distribution is logged and, if the distribution originator requested notification, a notification is sent to the originator or to the report destination specified in the distribution.

Note: System Network Architecture distribution services (SNADS) status distributions and distribution reports are used to report information about a distribution back to the originator. Status report distributions never result in another status report distribution. If a status report distribution is deleted, no notification is sent.

***PURGE**
Distributions on the queue are deleted. Deleted distributions are not logged and no notification is sent to the originator or to the report destination specified in the distribution.

Top

Examples

Example 1: Initializing a Distribution Queue

```
INZDSTQ   DSTQ('SYSTEMA APPN')
```

Connection information is about to be changed for system 'SYSTEMA APPN' by a central site administrator. This command initializes the queue to avoid error conditions that can be encountered by the Change Distribution Queue (CHGDSTQ) command. Distributions on the queue are not deleted.

Example 2: Initializing and Clearing a Distribution Queue

```
INZDSTQ   DSTQ('ERRORQ')  CLEAR(*YES)
```

This command clears the distribution queue ERRORQ that is being used as a repository for distributions that would have resulted in routing errors. Distributions that are deleted are logged, and the originators of the distributions are notified.

Example 3: Initializing and Purging a Distribution Queue

```
INZDSTQ   DSTQ('TESTQ')  CLEAR(*PURGE)
```

This command clears the distribution queue TESTQ that is being used for testing a new batch application. Distributions are deleted but not logged, and the originators are not notified.

Top

Error messages

***ESCAPE Messages**

CPF8802
Distribution queue &1 was not found.

CPF8807
Error occurred while using QSNADS journal.

CPF8809
Errors detected on SNADS internal queues.

CPF8812
Error occurred while processing distribution queues.

CPF8849
Queue &1 in use by another distribution services function.

CPF9845
Error occurred while opening file &1.

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

Top

Initialize NWS Configuration (INZNWSCFG)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Initialize NWS Configuration (INZNWSCFG) command initializes or resets a service processor (*SRVPRC) network server configuration when various service processor parameters are changed or need to be enabled.

Restrictions:

- This command is shipped with public exclude (*EXCLUDE) authority. When this command is shipped, authority is issued only to the security officer. The security officer can grant the use of this command to other users.
- You must have input/output system configuration (*IOSYSCFG) and security administrator (*SECADM) special authorities to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
NWSCFG	Network server configuration	*Communications name*	Required, Positional 1
OPTION	Processing option	*INIT, *CHGSPAUT, *REGEN, *SYNC	Required, Positional 2
SPAUT	SP authentication	Single values: *DFT Other values: *Element list*	Required, Positional 3
	Element 1: User name	*Character value*	
	Element 2: User password	*Character value*	

Top

Network server configuration (NWSCFG)

Specifies the name of the network server configuration.

This *NWSCFG object must have been created with *SRVPRC specified for the **Configuration type (TYPE)** parameter on the Create NWS Configuration (CRTNWSCFG) command.

This is a required parameter.

name Specify the service processor network server configuration to be processed.

Top

Processing option (OPTION)

Specifies option to perform with the service processor network server configuration.

This is a required parameter.

***INIT** Initialize a new service processor.

Note: This option is used to configure a service processor that has never been configured before such as when a new system is put into service or has had a service processor field replacement performed.

Note: A new user name and password must be specified on the **SP authentication (SPAUT)** parameter unless INZSP(*NONE) was specified in the corresponding **CRTNWSCFG** command

***CHGSPAUT**
Change the service processor user name and password to the new values specified on the **SP authentication (SPAUT)** parameter.

***REGEN**
Request that the service processor regenerate a self signed certificate. This option is used if the service processor's certificate has expired, or if a new certificate and password are desired at any time before the certificate expires.

Note: This option is only valid when INZSP(*AUTO) is specified in the corresponding **CRTNWSCFG** command.

Note: A new password must be specified on the **SP authentication (SPAUT)** parameter unless INZSP(*NONE) was specified in the corresponding **CRTNWSCFG** command

***SYNC**

Synchronize the user name and password and self-signed certificate from the service processor. This option is used to initially synchronize i5/OS with the service processor. It is also used if multiple service processor network server configurations are used for the same system or the service processor network server configuration has been restored from backup and the service processor certificate must be synchronized. The current user name and password for the service processor must be specified on the **SP authentication (SPAUT)** parameter to perform this option.

Top

SP authentication (SPAUT)

Specifies the service processor user name and password. This is used to authenticate and secure the service processor.

Note: The *DFT value is only valid when INZSP(*NONE) was specified in the corresponding **CRTNWSCFG** command.

This is a required parameter.

Single values

***DFT** The default service processor userid and password are used.

Element 1: User name

character-value
Specify a name that represents the host configuration that owns the service processor. It is suggested that the remote system network server configuration name be used. If multiple remote

system network server configurations use the same service processor at different times, each configuration must contain the same user name and password.

Element 2: User password

character-value

Specify the service processor password. Password must be at least 5 characters in length and contain at least one alphabetic character and one numeric or symbolic character.

Top

Examples

Example 1: Regenerate Service Processor Certificate

```
INZNWSCFG NWSCFG(MYCONFIG)
          OPTION(*REGEN)
          SPAUT(username password)
```

This command regenerates the remote system's service processor certificate then changes the user name and password.

Example 2: Change Service Processor User Name and Password

```
INZNWSCFG NWSCFG(MYCONFIG)
          OPTION(*CHGSPAUT)
          SPAUT(username password)
```

This command changes the user name and password used to secure the service processor.

Top

Error messages

***ESCAPE Messages**

CPF0910

Password not valid for system.

CPF96CB

Network server configuration &1 not found.

CPF96CD

Network server configuration type &2 is not valid.

CPF96CE

Password matches the previous value.

CPF96CF

Errors processing network server configuration &1.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9899

Error occurred during processing of command.

Top

Initialize Optical (INZOPT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Initialize Optical (INZOPT) command initializes an optical volume. Depending on the type of optical volume being initialized this operation may take up to 30 minutes to complete. When an existing optical volume is initialized a second time, all existing information is lost.

Restriction: To use this command you must have *ALL authority to the authorization list securing the volume if it is in an optical media library device. You need *CHANGE authority to the authorization list securing the volume if it is in an optical device.

Top

Parameters

Keyword	Description	Choices	Notes
VOL	Volume identifier	*Character value*, ***MOUNTED**	Optional, Key, Positional 1
NEWVOL	New volume identifier	*Character value*, ***VOL**	Optional, Positional 2
DEV	Device	*Name*	Optional
THRESHOLD	Volume full threshold	1-100, ***CALC**	Optional
CHECK	Check for an active volume	*NO, ***YES**	Optional
ENDOPT	End of media option	***LEAVE**, *UNLOAD	Optional
CLEAR	Clear	***NO**, *YES	Optional
TEXT	Text ’description’	*Character value*, ***BLANK**	Optional
TYPE	Volume type	***PRIMARY**, *BACKUP	Optional
CCSID	Coded character set ID	***CALC**, 500, 850	Optional
MEDFMT	Media format	***MEDTYPE**, *HPOFS, *UDF	Optional

Top

Volume identifier (VOL)

Specifies the volume identifier of the optical volume being initialized.

***MOUNTED**
The volume mounted in the specified device (DEV parameter) will be initialized.

volume-identifier
Specify the identifier of the optical volume to initialize.

Top

New volume identifier (NEWVOL)

Specifies the identifier of the optical volume after it is initialized. The identifier must contain only alphabetic characters (A through Z), numeric characters (0 through 9), hyphen (-), underscore (_), or a period (.). The first character must be alphabetic or numeric and the identifier cannot contain blanks.

***VOL** The new volume identifier is the same as the old volume identifier.

new-volume-identifier
Specify the new volume identifier.

Top

Device (DEV)

Specifies the optical device which contains the volume to be initialized. This parameter is only required when VOL(*MOUNTED) is specified. The device cannot be an optical media library device.

optical-device
The name of the optical device containing the volume which will be initialized.

Top

Volume full threshold (THRESHOLD)

Specifies the percentage of space on the volume to use until the volume is considered full. This field is only used if the media format is *HPOFS. For other media formats, this field is ignored and the threshold will default to 100 percent.

***CALC**
The system will calculate the percentage of the volume to use based on media format and volume type.

- For a media format of *HPOFS and a volume type of *PRIMARY the threshold will be 95 percent.
- For a media format of *HPOFS and a volume type of *BACKUP the threshold will be 99 percent.
- For a media format of *UDF the threshold will be 100 percent.

volume-full-threshold
Specify the volume threshold percentage. Valid values range from 1 through 100.

Note: If the volume type is *BACKUP, this parameter is ignored and the volume-full-threshold is set to 99 percent.

If the media format is *UDF, this parameter is ignored and the volume-full-threshold is set to 100 percent.

Top

Check for an active volume (CHECK)

Specifies whether the system checks to see if the optical volume is initialized.

***YES** The system checks to see if the optical volume is initialized. If the volume is initialized, the operation is ended and an error message is sent.

***NO** The system does not check to see if the optical volume is initialized. The volume will be initialized and all existing data will be lost.

Top

End of media option (ENDOPT)

Specifies whether the media is unloaded from the device after the initialize command completes.

Note: This parameter is ignored if the media is an optical library device.

***LEAVE**
When the initialize completes the media is left in the device.

***UNLOAD**
When the initialize completes the media is unloaded from the device.

Top

Clear (CLEAR)

Specifies whether or not existing data on the volume will be cleared during the initialize process. This parameter only applies when the volume media type is *DVD-RAM.

Note: If the volume media type is *WORM the volume is never cleared regardless of the parameter setting.

If the volume media type is *ERASE the volume is always cleared regardless of the parameter setting.

***NO** The volume is not cleared.

***YES** The volume is cleared of existing data prior to initialization.

Note: If this option is selected the INZOPT command may take several hours or more to complete, depending on the media capacity.

Top

Volume type (TEXT)

Specifies the text that briefly describes the optical volume.

***BLANK**
Text is not specified.

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

Top

Volume type (TYPE)

Specifies the type of optical volume being initialized. Optical volumes for user applications are initialized as primary volumes. Backup optical volumes can be written to only by using the following set of optical backup commands: CVTOPTBKU, CPYOPT, and DUPOPT.

***PRIMARY**
The optical volume is used as a primary volume.

***BACKUP**
The optical volume is used as a backup volume.

Top

Coded character set ID (CCSID)

Specifies the character set in which the optical volume, directory, file names, and volume description are written. This parameter does not affect how user data is written. The user application must determine the character set in which the file data is written.

***CALC**
The system will select the default character set based on the media format.

500 The EBCDIC character set and code page 500 are used.

850 The ASCII character set and code page 850 are used.

Top

Media format (MEDFMT)

Indicates the media format to use when writing to the optical media. There are two media formats, either *HPOFS (High Performance Optical File System) or *UDF (Universal Disk Format). For a complete comparison of the two media formats refer to the Optical Support, SC41-4310 book.

***MEDTYPE**
Specifies that the operating system will determine which media format is used to initialize the volume. The media format will be based upon the media type.

- If the media type is *WORM or *UNKNOWN, the media will be initialized using the *HPOFS format.
- If the media type is *ERASE and has not been previously initialized the media will be initialized using the *HPOFS format.
- If the media type is *ERASE and has been previously initialized it will be initialized using the previous media format.
- If the media type is *DVD-RAM, the media will be initialized using the *UDF format.

***HPOFS**
The High Performance Optical File System is used to initialize the volume. One of the characteristics of HPOFS is space occupied by a deleted file is not reused. The only way deleted file space can be recovered is to reinitialize the media thereby losing all previously recorded data on the media.

***UDF** The Universal Disk Format, a subset of the ISO 13346 standard, is used to initialize the volume. One of the characteristics of UDF is space occupied by a deleted file will be reused when needed for either the creation of a new file or the extension of an existing file. The UDF media format also provides file and directory level security through the use of permissions.

Top

Examples

```
INZOPT   VOL(VOL01)  THRESHOLD(99)  CHECK(*NO)
```

This command initializes the optical volume VOL01 with a volume-full-threshold of 99 percent. The system does not check to see if the volume is initialized.

Top

Error messages

***ESCAPE Messages**

OPT1305
Optical volume &1 is read only.

OPT1315
Optical volume &1 is write protected.

OPT1320
Optical volume &1 in use.

OPT1325
Optical volume format not recognized.

OPT1330
Optical volume not found or not useable.

OPT1331
Optical volume &1 not found.

OPT1335
Volume &1 already initialized.

OPT1342
Invalid volume identifier specified.

OPT1345
No free space available on media.

OPT1346
Operation not allowed to volume located in a remote optical device.

OPT1350
Write operation failed to optical volume &1.

OPT1360
Media directory corrupted on optical volume &1.

OPT1375
Optical volume &1 already exists.

OPT1460
Optical volume &1 is not in an optical device.

OPT1485
Initialize or rename of optical volume failed.

OPT1489
Volume parameter is not permitted for device &1.

OPT1530
&1 does not represent a valid optical device.

OPT1540
Invalid parameters specified.

OPT1555
Optical device &1 in use.

OPT1605
Media or device error occurred.

OPT1790
Operation not allowed or conflicts with another request.

OPT1805
Error accessing optical volume index file.

OPT1810
Error accessing optical directory index file.

OPT1815
Internal program error occurred.

OPT1820
Internal error occurred on optical device &1.

OPT1821
Error occurred on optical device &1.

OPT1825
Optical indexes are incorrect for optical device &1.

OPT1860
Request to optical device &1 failed.

OPT1861
No device description configured for resource &1.

OPT1862
No active device description for resource &1.

OPT1863
Optical libraries need to be reclaimed.

OPT1872
Optical request timed out or was cancelled.

OPT2301
Internal system object in use.

OPT2420
Not authorized to optical volume &2.

OPT2422
Not authorized to file or directory.

OPT7740
User not authorized to object &2 in library &3 type &4.

Top

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This i5/OS commands publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
Distributed Relational Database Architecture
Domino
DRDA
IBM
Infoprint
InfoWindow
i5/OS
iSeries
Integrated Language Environment
Lotus
LPDA
OfficeVision
Print Services Facility
RPG/400
System i
System x
SystemView
System/36
TCS
Tivoli
WebSphere
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Printed in USA