

North American Response Centers

HP 3000 APPLICATION NOTE #20

OUTPUT SPOOLING



January 1, 1987
Document P/N 5958-5824/2701

HP 3000 APPLICATION NOTES are published by the North American Response Centers twice a month and distributed with the Software Status Bulletin. These notes address topics, where the volume of calls received at the Centers indicates a need for addition to or consolidation of information available through HP support services. You may obtain previous notes (single copies only, please) by returning the attached Reader Comment Sheet listing their numbers.

| <u>Note #</u> | <u>Published</u> | <u>Topic</u> |
|---------------|------------------|--|
| 1 | 2/21/85 | HP 3000 Printer Configuration Guide (superseded by note #4) |
| 2 | 10/15/85 | Terminal Types for HP 3000 HPIB Computers (superseded by note #13) |
| 3 | 4/01/86 | HP 3000 Plotter Configuration Guide |
| 4 | 4/15/86 | HP 3000 Printer Configuration Guide - Revised |
| 5 | 5/01/86 | MPE System Logfile Record Formats |
| 6 | 5/15/86 | HP 3000 Stack Operation |
| 7 | 6/01/86 | Cobol II/3000 Programs: Tracing Illegal Data |
| 8 | 6/15/86 | KSAM Topics: Cobol's Index I/O; File Data Integrity |
| 9 | 7/01/86 | Port Failures, Terminal Hangs, TERMDSM |
| 10 | 7/15/86 | Serial Printers - Configuration, Cabling, Muxes |
| 11 | 8/01/86 | System Configuration or System Table Related Errors |
| 12 | 8/15/86 | Pascal/3000 - Using Dynamic Variables |
| 13 | 9/01/86 | Terminal Types for HP 3000 HPIB Computers - Revised |
| 14 | 9/15/86 | Laser Printers - A Software and Hardware Overview |
| 15 | 10/01/86 | FORTRAN Language Considerations - A Guide to Common Problems |
| 16 | 10/15/86 | IMAGE: Updating to TurboImage & Improving Data Base Loads |
| 17 | 11/01/86 | Optimizing VPLUS Utilization |
| 18 | 11/15/86 | The Case of the Suspect Track for 792X Disc Drives |
| 19 | 12/01/86 | Stack Overflows: Causes & Cures for COBOL II Programs |

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

This document contains proprietary information which is protected copyright. All rights are reserved. Permission to copy all or part of this document is granted provided that the copies are not made or distributed for direct commercial advantage; that this copyright notice, and the title of the publication and its date appear; and that notice is given that copying is by permission of Hewlett-Packard Company. To copy otherwise, or to republish, requires prior written consent of Hewlett-Packard Company.

Output Spooling

"SPOOL" is an acronym for System Peripheral Output On Line. It provides the illusion of "sharing" devices, such as card readers and printers, among many users simultaneously. By temporarily storing data, coming from or destined for these devices, each user can perform I/O without concern for device ownership.

Spooling has two forms: input spooling and output spooling.

Examples of input spooling devices are card readers, tape drives and terminals. MPE optionally allows input spoolfiles to be created from disc with the STREAM command. Input spooling also controls the queuing and managing of job streams.

Output, however, is by far the most common usage of spooling. This note will discuss output spooling by giving an overview of the spooling process, describing the commands and utilities which allow users and operators to control the spooling process, and finally, discussing some of the more common output spooling problems which may be encountered. This note will not address configuration of printers or SPOOLEE I/O errors as those topics are covered in Application Note #4, *HP 3000 Printer Configuration Guide*; Application Note #10, *Serial Printers - Configuration, Cabling, Muxes*; and the May 15, 1986 issue of *Response Center Questions & Answers*.

How Output Spooling Works

Overview

Output spooling is divided into two independent functions. The first function accepts data intended for a spooled printer and creates a spoolfile. The second function takes an existing spoolfile, and performs the actual output to the printer.

In order for MPE to create a spoolfile, a printer must be spooled (i.e the spool queue must be open), and enough disc space must be available to contain the output.

In order for MPE to print the spoolfile, the printer must be available, and a spooler process must be running for the port to which the printer is attached. A *Spooler Process*, created by a :STARTSPOOL *ldev* command, selects and prints an already existing spoolfile. Each spooled printer has its own spooler process that runs independently of any other spooler process.

MPE determines whether or not a device is spooled depending on the device's I/O configuration and *whether or not the spool queue for that device is opened*. A spool queue is a flag set for a device or class to signify that output is to be sent to a spoolfile rather than the device. If the queue is opened for a device or class, then the file is spooled. The queue does not affect the printing of spoolfiles. Therefore, a spool queue can be open and spoolfiles can be created even if there is no spooler process to print them. The spooler queue is opened by the :STARTSPOOL *class* or by the OPENQ command.

Only a logical device can have a spooler process associated with it. *It is important to note that only an LDEV can have a spooler process, not a "spooled CLASS"*. A spooled class is used to logically link more than one printer in the output process so that a user can print to any of several printers without concern as to each printer status or load. (For example, "LP" is a common spooled class.)

A spooled class is a class that has a spool queue open. The term is misleading since a spooled class does not have a spooler process. *In order to print out spoolfiles sent to the class, one or more of the LDEV's in the class must have a spooler process.*

Spoolfiles are special disc files that can only reside on discs in the device class SPOOL. They are considered special because:

- 1) Spoolfiles have their own directories which are not part of the permanent disc directory. Instead, they are maintained in memory data segments. These segments can potentially be damaged by a System Failure. For this reason they are rebuilt at system startup unless a WARMSTART is done.

During a WARMSTART, spoolfiles are not rebuilt but are retained on disc and can be stored to tape. Because it's possible that memory structures corrupted during the failure can be carried over in a WARMSTART, it is always recommended that you cold load the system after you have saved your spoolfiles.

- 2) Spoolfiles also have a special variable length record structure that is a modification of regular variable length records.

Creating a Spoolfile

When your program opens a file, the FOPEN intrinsic is called to do the actual work. If the output is directed towards a non-sharable device, such as a printer, the FOPEN code determines whether a spoolfile needs to be created or not. How it decides depends on whether the FOPEN call requested a particular LDEV or just any LDEV in a certain device class.

If a LDEV has been specified in the FOPEN intrinsic call, then a check is made to see if the LDEV's spool queue is open. If the queue is opened, then a spoolfile needs to be created.

If the queue is not open, FOPEN will check if the device is available (not in use by another process), up (not taken out of service by a :DOWN command), and "real" (not a virtual type device). If so, then the printer is opened and run "HOT". If not, FOPEN will fail FSERR 55, DEVICE UNAVAILABLE.

If a device class has been specified in the FOPEN call, a check is made to see if the device class spool queue is open. If so, a spoolfile needs to be created.

If the device class queue is not open, and *only one copy of the output is needed*, MPE will search, starting with the first LDEV in the class, for an available, up, and real device. Upon finding one it will open it "HOT". If no such device can be found, a search will be made for any spooled LDEV in the class. If one is found, then a spoolfile needs to be created. Otherwise, FOPEN will fail with FSERR 55, DEVICE UNAVAILABLE.

If the device class queue is not open, and *more than one copy of the output is needed*, MPE will search for any spooled LDEV in the class. If one is found, then a spoolfile can be created. If not, a search will be made for the first available, up, and "real" LDEV. If one is found, the number of copies to print is reduced to one and the device is opened "HOT". Otherwise, FOPEN will fail with FSERR 55, DEVICE UNAVAILABLE.

If a spoolfile is to be created, FOPEN will choose a disc on which to locate it. The disc will be chosen, using a round robin scheme, from those discs in the device class SPOOL. Then, the internal directory and virtual LDEV assignments are made for the spoolfile and it becomes available to your program for writing. Following the close of the spoolfile, any unused space in its last extent will be returned to the system.

NOTE: The HP 2680 and 2688 laser printers can only be run spooled. d 334

Printing a Spoolfile

A spooling process is started by a STARTSPOOL *ldev* command. If you have configured a device as initially spooled, this will occur automatically whenever the system is restarted (except with a WARMSTART).

Once a spoolfile is created, it is linked into a list of ready spoolfiles ordered by priority, class, and time of creation. The spooler process then selects and prints one of the files. If no files are ready to print, then the spooler process hibernates. By altering or creating a spoolfile, the spooler process for *every* printer wakes up and scans the list to see if there is a file it should print.

Note: when a spooler process is started, it creates a list of every class that contains its LDEV. The spooler will thus print any file that is sent to its LDEV or any file that was sent to a class that it shares. If more than one spooler process exist for the same class, then the first spooler to select the spoolfile will print it.

The spooler selects files to be printed by the following algorithm:

- 1) Check if there are any previously active files that are ready. This is used, for example, when a RESUMESPOOL,BEGINNING is issued. Note: a normal RESUMESPOOL will not leave the spoolfile in a previous active state, thus it will not continue printing if there is a higher priority file waiting.
- 2) If there are no previously active files then the spoolfile with the highest priority for this LDEV or device class to which the LDEV is a member will be printed. If there are spoolfiles with equal priorities in the class chain and the LDEV chain, then the spooler will alternate between the two chains.

In the selection process, both the LDEV chain and the class chain are scanned for files. The LDEV chain contains all spoolfiles that have the LDEV of the spooler, and the class chain contains all spoolfiles that have a class that the spooler LDEV is included in. There is only one class chain for the entire system. Both the LDEV and the CLASS chains are displayed with the SHOWOUT SP command.

- 3) If a device belongs to two or more classes, its spooling process will print spoolfiles in one class and then those of equal priority in other classes.
- 4) If there are still two or more files of the same priority, then the oldest file will be selected first.

When a spoolfile is printed which is linked in a class chain, the file is temporarily moved into the LDEV chain, and assigned the LDEV of the spooler that selected it. If the file then goes ready again, it is relinked into the class chain. This is important if more than one copy is to be printed. After a copy is printed, the count is updated, and the spoolfile will be relinked into the class chain to be selected and printed again. If a new file of the same priority was made ready in this time period, it will print next rather than the multiple copy spoolfile. This is a result of a new time stamp being put on the old file when it is relinked into the class chain.

Configuring Your System For Spooling

You can configure your system for spooling by using the SYSDUMP command or in the INITIAL dialog during system startup. The parameters involved are discussed below.

Configuration Parameters

I/O CONFIGURATION CHANGES. Spoolfiles can only exist on disc that are configured with a device class "SPOOL". Note, however, that adding a disc to device class SPOOL does not reserve space for spoolfiles on it; it only allows for them to be created on that disc. If a spoolfile needs to be created and space is not available on any discs in the device class SPOOL, the spool queues will be shut and you will receive the console message "NO MORE SPACE IN SPOOL CLASS, ALL QUEUES SHUT."

MAX NUMBER OF OPEN SPOOLFILES. This parameter determines the size of certain internal tables, and sets the number of virtual LDEV's that are available. A virtual LDEV is not a real logical device. It is a dummy parameter which allows the MPE file system to handle spoolfiles as if they were normal MPE files. Each open spoolfile requires one such virtual LDEV.

MAX NUMBER OF SPOOLFILES KILOSECTORS. This parameter specifies the total amount of spoolfile space that can be in use at one time. Since spoolfile space is not reserved, but is obtained as needed, this can be configured large at no real cost. If this limit is reached, you will receive the console message "MAX SPOOLFILE KILOSECTORS IN USE, ALL QUEUES SHUT" and the spool queues will be shut.

NUMBER OF SECTORS PER SPOOLFILE EXTENT. Since a spoolfile can only have 32 extents, this parameter essentially limits its size. Because spoolfile records are variable in length and have their trailing blanks stripped, the size of a spoolfile is very dependent on the data contained in it. This makes it difficult to determine the number of sectors a spoolfile will actually need. As a rule of thumb, however, a 500 page report will require about 384 sectors.

If you make this parameter too large and your disc space is badly fragmented, (i.e. there are no large contiguous sections), the spooler will be unable to find an extent large enough will shut the queues with a "NO MORE SPACE IN SPOOL CLASS, ALL QUEUES SHUT" console message.

Spooling Related MPE Commands

There are several commands which are available to control spooling. A brief description of each is shown below. More detailed information is provided in the *MPE V Commands Reference Manual (P/N 32033-90006)*, the *System Operation and Resource Management Reference Manual (P/N 32033-90005)*, and the *MPE IV Console Operator's Guide (P/N 32002-90004)*.

In the following commands, it is important to note that use of the CLASS parameter has a very different meaning from the LDEV parameter. The LDEV parameter alters a spooling process and the CLASS parameter deals with only the state of the class queue.

STARTSPOOL {*ldev*[:SHUTQ]}
 {*devclass*}

STARTSPOOL has two forms: STARTSPOOL specifying a device class, and STARTSPOOL specifying a LDEV. If you use a device class, the queue for that class is opened allowing spoolfiles to be created, but NO spooler processes are created to print them. If you use a LDEV, the queue for that LDEV is opened (unless you use the optional SHUTQ parameter),

and a spooler process is created. The device classes with which the LDEV is associated are unaffected, however.

If you have configured the printer as initially spooled, MPE does an implicit STARTSPOOL LDEV command at system startup.

STOPSPPOOL $\left\{ \begin{array}{l} ldev[;OPENQ] \\ devclass \end{array} \right\}$

STOPSPPOOL also has two forms. STOPSPPOOL specifying a LDEV stops the spooler process and shuts the queue for the LDEV. If the OPENQ parameter is used, the spooler process is stopped but the queue is kept open. STOPSPPOOL specifying a device class shuts the queue for that device class without affecting the spooler processes or queues for LDEVs in the class.

OPENQ $\left\{ \begin{array}{l} ldev \\ devclass \end{array} \right\}$

The OPENQ command is only available on MPE versions V/E (G.00.00) or later and is the same as the OPENQ parameter of the STOPSPPOOL command. It opens the queue for a LDEV or a class but does not affect any spooler processes.

SHUTQ $\left\{ \begin{array}{l} ldev \\ devclass \end{array} \right\}$

SHUTQ is also only available on MPE versions V/E or later and is similar to the SHUTQ parameter of STARTSPOOL. It shuts the queue for a LDEV or class so that no spoolfiles can be created for output sent to the class or LDEV.

OUTFENCE *priority*[;LDEV=*ldev*]

OUTFENCE is used to determine which spool files can be printed by a spooler process. The spoolfile's priority must be higher than the current setting of OUTFENCE (for that LDEV) for the spooler process to print it.

SUSPENDSPOOL *ldev*[;FINISH]

SUSPENDSPOOL suspends the spooler process for the specified LDEV and returns any ACTIVE spoolfile to the READY state. If the FINISH parameter is specified, the current file will be printed first.

SUSPENDSPOOL does not delete any part of a spoolfile. As with the STOPSPPOOL command, if a file is being printed and this command is used, the file is still complete.

RESUMESPOOL *ldev* $\left[\begin{array}{l} ;BACK \quad [nnn \text{ FILES}] \\ ;FORWARD [nnn \text{ PAGES}] \\ ;BEGINNING \end{array} \right]$

RESUMESPOOL resumes a spooler process if it has been suspended. If none of the optional parameters are specified, printing will resume with the highest priority spoolfile. This may or may not be the spoolfile that was printing when the process was suspended. To ensure that the active file is resumed, you should use the BACK, FORWARD, or BEGINNING parameters.

ALTSPOOLFILE {#Onnn} {ldev} {;PRI=outpriority
;COPIES=numcopies
;DEFER
;DEV={ldev
devclass}}

ALTSPOOLFILE allows you to change the priority, number of copies, and destination device of a spoolfile. If the device is changed, the spoolfile will not print unless the new device has a spooler process. By changing the device, a spoolfile can be sent to a device that has its queue shut.

If an LDEV is specified in the command, then the current active spoolfile for the device will be changed.

DELETESPOOLFILE {#Innn} {ldev} {#Onnn}

DELETESPOOLFILE deletes a spoolfile that is in either the ACTIVE or READY state. If the LDEV option is used, then the current active spoolfile for the device will be deleted.

HEADON ldev

HEADON instructs the spooler process for the specified LDEV to print headers (banner pages for each spoolfile containing information on the job or session that created it).

MPE does an implicit HEADON for each LDEV at system startup.

HEADOFF ldev

HEADOFF suppresses the printing of headers by the spooler process of that LDEV.

SHOWOUT SP

SHOWOUT (shown here with only its SP parameter) displays information on the status of spoolfiles as follows:

ACTIVE: The spoolfile is currently being printed.

OPENED: The spoolfile has been opened for write access by a program.

READY: The spoolfile is closed and is in the queue waiting to be printed.

LOCKED: The spoolfile was in the READY state, but is currently opened by the system or a program (most likely the SPOOK utility).

SHOWDEV {ldev} {devclass}

SHOWDEV displays information on the status of a LDEV or the LDEVs in a particular device class. This command is very useful in helping you resolve spooling problems. (Refer to the *Handling Spooling Difficulties* section at the end of this note.)

The AVAIL column indicates the status of the device. "AVAIL" found here means the LDEV is available for spooling or allocation to another process. "UNAVAIL" means that the LDEV is opened exclusively by some process. (This could be a spooler process.) "SPOOLED" means that the spool queue for this LDEV is open and spoolfiles can be created for it.

The OWNERSHIP column signifies who owns the LDEV. If a spooler process exists for this LDEV, "SPOOLER OUT" will appear here. If the LDEV is run "HOT", the job or session number will appear.

It is important to remember that SHOWDEV only displays information regarding the devices and *not* the spool queues. In particular, note that LDEVs in a device class for which a STARTSPOOL has been issued will still be shown as AVAIL since no spooler processes were created by that STARTSPOOL.

Also note that SHOWDEV does not indicate if a spooler process has been suspended or not. For this reason, if it appears that the spooler is not running you should attempt a RESUMESPOOL command on the chance that SUSPENDSPOOL was done earlier.

The following shows the output display from SHOWDEV and how various commands affect the state of the LDEV:

| Commands | SHOWDEV Output | | |
|---------------|----------------|---------|-------------|
| STOPSPPOOL 18 | | | |
| SHUTQ 18 | | | |
| SHOWDEV 18 | LDEV | AVAIL | OWNERSHIP |
| | 18 | AVAIL | |
| STARTSPOOL 18 | | | |
| SHOWDEV 18 | LDEV | AVAIL | OWNERSHIP |
| | 18 | SPOOLED | SPOOLER OUT |
| STARTSPOOL 18 | | | |
| SHUTQ 18 | | | |
| SHOWDEV 18 | LDEV | AVAIL | OWNERSHIP |
| | 18 | UNAVAIL | SPOOLER OUT |
| STOPSPPOOL 18 | | | |
| OPENQ 18 | | | |
| SHOWDEV 18 | LDEV | AVAIL | OWNERSHIP |
| | 18 | SPOOLED | |

SPOOK – The Spoolfile Utility Program

SPOOK is a program that manipulates spoolfiles. It provides many of the same functions as the ALTSPoolFILE command. In addition, it allows you to examine spoolfiles, copy them, or store them on tape for later retrieval or transferral to another system. Its use is documented in the *MPE V System Utilities Reference Manual (P/N 32033-90008)* and the *MPE IV System Utilities Reference Manual (P/N 32033-90044)*.

Three different versions of the program exist: SPOOK, SPOOK4, and SPOOK5. The programs all perform the same functions but are designed for use with different versions of MPE. It is very important that you use the version of SPOOK that matches your version of MPE. Failure to do so may result in a System Failure.

The original version of the utility is named SPOOK and can only be run on MPE IV, V/P, or V/P Delta-1 systems. SPOOK can only read tapes made by other SPOOK programs on MPE IV systems.

SPOOK5 replaces the original SPOOK for MPE V systems because MPE V's internal tables containing spooling information differ in format from those on MPE IV systems. SPOOK5, however, can read spook tapes generated on either MPE IV or V systems.

SPOOK4 runs only on MPE IV, V/P, or V/P Delta-1 systems and allows them to read SPOOK5 tapes created on MPE V systems.

Handling Spooling Difficulties

This section discusses some of the common spooling problems you may encounter and provides assistance in dealing with them.

UNAVIL SYS

If SHOWDEV shows a supposedly spooled LDEV as unavail sys, then one of the following may be happening:

- 1) If a device is owned by MPE, and if the spooler queue is shut, then SHOWDEV will display the device as being unavailable.
- 2) A SHOWDEV will also indicate unavail sys if a spooling process has aborted, but user I/O is still pending to its device. Because the spooler aborted, many of the MPE I/O linkages have been cleaned up such that the I/O cannot proceed. Some things you can try to clear up this situation are:
 - Issue a RECALL, and make sure that no requests are pending for the LDEV. If there are, issue a REPLY pin,0.
 - Issue an ABORTIO ldev until the message appears that there is no more I/O to abort.

- Cycle the power to the printer.
- Make sure that the address of the printer has not changed, and that the cables are all connected.

If all the above fail, and the printer is on an ATP or ADCC, TERMDSM reset can be used. As a last resort, a WARMSTART should clear the printer.

SPOOLER BUSY, TRY AGAIN

Any time a spooler command is issued, the command is passed to the spooler through a word on the spooler's stack. Once a command has been issued, a flag is set indicating a new command is entered. If another command is sent, but the spooler has not had time to process the previous command (i.e. reset the flag), then the 'busy' message is displayed. Only the spooler can reset this flag.

This is not a planned state, but can occur on a heavily loaded system where a RESUMESPOOL; BACK nn is issued. In this case, wait for the spooler to finish processing the backlog of commands.

Another case where this error can occur is when the spooler is reading, writing, or aborting I/O and receives an unrecognized status code. If it does, then the spooler will loop and try again until it gets a recognized status. Note that an error status is a recognizable status code.

The following list are cases where this can happen:

- 1) Power fail restart
- 2) 2680 I/O errors will cause status checks.
- 3) Printer sends a bad status while a write is in progress.
- 4) Printer sends a bad status while an I/O is being aborted by the spooler.

Note: if the status is readable, then the spooler will return a spooler I/O error.

If this error is returned, and waiting does not prevail, then try the following:

- 1) Issue a RECALL to see if any replies are pending. This is the most common cause of the spooler busy message.
- 2) Issue an ABORTIO to the LDEV until you receive a message which indicates there is no more I/O to abort. Since this is an external request, it will bypass the spooler's I/O abort.
- 3) Cycle power to the printer, and make sure that the cables are connected properly. Also, perform a SHOWDEV on the device to check what state it is in.
- 4) If the printer is on an ATP or ADCC, try a TERMDSM reset. If this fails, then a WARMSTART should clear it up.

JOB OVERLOAD TYPE 6 (UNABLE TO ALLOCATE \$STDLIST):

If the spooler runs out of spool space, it will shut the queue so that no more spoolfiles can be created. after the files are printed, the operator must open the queue back up.

The above error will occur when a job stream tries to create a spoolfile for its standard output. Please consult the April 15, 1986 issue of *Response Center Questions & Answers* for a complete explanation of this error.

