HP 3000 MPE/iX Computer Systems

Native Mode Spooler Reference Manual

HP Part No. 32650-90166
Printed in U.S.A. September 1998

Edition 5
E0998

PostScript® is a registered trademark of Adobe Systems, Inc.

UNIX® is a registered trademark of Unix System Laboratories, Inc. in the U.S.A. and other countries.

Acknowledgements

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Restricted Rights Legend

Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time that this document was issued. Many product releases do not require changes to the document; therefore, do not expect a one-to-one correspondence between product releases and document editions.

Edition	Date	Software Version
First Edition	April 1990	A.40.00
Second Edition	June 1992	B.40.00
Third Edition	April 1994	C.50.00
Fourth Edition	July 1996	C.55.00
Fifth Edition	September 1998	C.60.00

Preface

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

You may encounter references to MPE V, which is the operating system for HP 3000s, not based on PA-RISC architecture. MPE V software can be run on the PA-RISC (Series 900) HP 3000s in what is known as *compatibility mode*.

This *Native Mode Spooler Reference Manual* (32650-90166) is written for the general user and the system manager. It describes how the standard MPE/iX user can view and manipulate spool files and it describes spooler management tasks for users with System Manager (SM) or System Supervisor (OP) capabilities.

The contents of this manual are:

Chapter 1 **Getting Started** contains introductory information about functions and commands, types of users, and hardware requirements and restrictions. It also demonstrates how to create, view, change, copy, save and delete a spool file.

Chapter 2 **Spooler and Spool File Management Tasks** describes how to manage the spooler and alter existing spool files.

Chapter 3 **Network Printer Operation** provides information for system managers on configuring and operating network printers, and information for all other users on customizing network printer output.

Chapter 4 **Commands Reference** contains commands and command syntax.

Chapter 5 **Utilities** describes the `SPIFF`, `SPFXFER`, and `PRINTSPF` utilities.

Appendix A **Spooler File Block Format (SBF)** provides the format of the spool file variable length records.

Appendix B **Spooler Command Comparison** compares native mode spooler commands with compatibility mode spooler commands.

Appendix C **Page Level Recovery and Checkpoints** provides basic information on what page level recovery and checkpoints are and how to use them.

Appendix D **Migration Information and Limitations** contains information about the changes involved in moving to the Native Mode Spooler and describes the limitations of NMS.

Glossary

Index

Contents

Figures

Tables

1

Getting Started

This chapter introduces you to the MPE/iX native mode spooler. It includes the following topics:

- a general description of the native mode spooler
- system requirements and restrictions
- a brief description of the spooler commands and utilities and their purpose
- working with spool files, including creating, saving, copying, altering and deleting a spool file
- viewing the output of a spool file

What is the native mode spooler?

The native mode spooler (NMS) is an MPE/iX subsystem used to manage and control print files and printing devices. *Spool* is an abbreviation for *simultaneous peripheral operation online*. A *spooler* allows numerous user processes requiring a printer to run simultaneously. This means that many programs may share a single printer.

Generally, users perform eight functions to control standard files on an MPE/iX system: create, list, print, alter, browse, copy, rename, and purge. The MPE/iX native mode spooler provides an easy, powerful way for you to perform the same functions with spool files because it has made them regular, visible files.

Starting with Release 5.5 of MPE/iX, you can use the spooler to control printing on devices that are directly attached to an HP 3000 as well as on devices that are attached via a network. Most of the information in this manual pertains to both non-network and network printers. Chapter 3 deals exclusively with network printing and explains the configuration and operation of network printers in detail.

System requirements and restrictions

The native mode spooler is part of the fundamental operating system (FOS) which you install on the HP 3000 with the `INSTALL` or `UPDATE` utility. It runs on any Hewlett-Packard Precision Architecture HP 3000 systems that have been updated to MPE/iX version A.40.00 (release 2.1 or later). Network printing requires Release 5.5 or later of MPE/iX.

A note on device configuration

Your system manager is responsible for properly configuring the output devices that are managed by the native mode spooler program. Configuring channel-attached printers or plotters using the NMMGR utility is not explained in this manual. For information, refer to the *System Startup, Configuration, and Shutdown Reference Manual* (32650-90042), the *HP 3000/iX Network Planning and Configuration Guide* (36922-61023) and the *Configuring Systems for Terminals, Printers, and Other Serial Devices* (32022-61000). To configure a network printer, read Chapter 3 in this manual.

Spooler commands, utilities, and user capabilities

To control spool files and spooled devices, you use an assortment of MPE/iX command and three utilities, SPIFF, SPFXFER and PRINTSPF. Each of these commands and utilities is briefly described in the table below.

The degree of control you have over the spooler and spool files depends primarily on the user capabilities you have been assigned, or whether or not you have access to the console. The four different levels of capability include system manager (SM), operator (OP), account manager (AM), and general user capabilities. The spooler commands behave differently and display different information depending upon the capabilities of the person issuing a command. Chapter 2, which explains spooler management tasks, points out such differences.

Table 1-1. Summary of Spooler Commands and Utilities

Command	Definition
BUILD	Creates and immediately allocates a new empty file on disk. You may use the *filecode* parameter of this command to specify the type of file. Three of these codes, appropriate to spool files, have been reserved, one each to designate an output spool file, an input spool file, and a checkpoint file for an output spool file. You may also use the SPOOL option of the BUILD command to create an output spool file that is not linked to the spool file directory SPFDIR.
COPY	Copies a nonprivate output spool file to another, new file which is not linked to the spool file directory SPFDIR.
FILE	Declares the file attributes to be used when a file is opened, which may override programmatic or system default file specificiations. Use the FILE command to declare the type of file, to create an output spool file that is not linked to the spool file directory, to mount special forms, to declare a spool file private, and to have the spooler save an output spool file after all copies have been printed.
FORMSALIGN	Initiates a forms message dialog with the system operator when the current spool file includes a special forms message. You issue this command for a specific LDEV or a device class.
JOB	Defines a job and allows you to specify if the output spool file the job produces is private, and if it is saved after all copies have been printed.

Table 1-1. Summary of Spooler Commands and Utilities (continued)

Command	Definition
LISTEQ	Displays all active file equations for a job or session so that you can find out if a job's output spool file is private or will be saved.
LISTF	Displays a list of one or more files for the system, account or group. The spool files you see listed depend upon your capabilities. System managers (SM capability) can view all files on the system. To list input and output spool files, issue a LISTF for IN.HPSPOOL and OUT.HPSPOOL respectively.
LISTFILE	Displays a list of one or more files in hierarchical directories.
LISTSPF	Produces a listing of input and output spool files.
OPENQ	Opens the spool queue for a specifiedlogical device, device name, or all devices of a device class.
OUTFENCE	Defines the minimum priority that an output spool file must have in order to be printed.
PRINTSPF	Displays the data and the special overhead area of each record of a spool file.
PURGE	Deletes a file (including a spool file) from the system.
RENAME	Allows you to change the identity (file name, lockword, and/or group name) of a spool file to which you have access and that is not linked to the spool file directory.
RESTORE	Copies nonprivate, native mode output spool files from backup media to an MPE/iX system that also has the native mode spooler.
SHOWDEV	Reports the status of a specific device or all devices in a single device class (such as LP).
SPFXFER	Transports spool files between MPE/iX systems that have the native mode spooler and MPE system that do not.
SPIFF	Allows you to list, manipulate, and transfer spool files.
SPOOLER	The general command you use to control spooler processes. This includes starting, stopping, suspending and resuming a spooler, and enabling and disabling spooling for a specified LDEV or device class
SPOOLF	Allows a qualified user to alter, print, or delete one or more output spool files.
STORE	Copies nonprivate, native mode output spool files to backup media for transfer (via the RESTORE command) to another MPE/iX system that also has the native mode spooler.

For detailed documentation on the commands, refer to chapter 4. For details on the utilities, refer to chapter 5.

Working with spool files

To the typical MPE/iX user, the Native Mode Spooler makes it appear as though he or she has exclusive access to the printer. That is, whenever users need to print something, or if they stream a job that produces a printed report, they simply issue a command without checking to see if the printer is busy or not and the spooler handles the rest.

The way that the spooler manages shared access to a single printer is by creating a spool file that contains, among other information, the text of the report. The spooler adds this file to the spool queue for the printer (if the queue is open) and then sends the spool file to the printer when it reaches the top of the queue. Once the report is printed, the spool file is deleted from the queue (unless you explicitly choose otherwise), which means that a list of spool files typically is equivalent to a list of files waiting to be printed.

This section explains the types of spool files and shows how MPE/iX users can work with their own spool files. (The spooler allows members of the system adminstration staff, depending upon their assigned capabilities, to manipulate all individual spool files in the queue and the spool queue itself. This chapter does not explain spool management.) In the next section, you'll learn the different ways to view spool files.

Types of spool files

With the native mode spooler (NMS), the file system creates spool files as ordinary, permanent disk files. There are two kinds of spool files it creates, input spool files, which are always linked to the spool file directory, and output spool files, which may or may not be linked to the spool file directory. The NMS also creates a third kind of disk file, checkpoint files, which are used to assist in the printing of output spool files.

When the spooler generates these files, it automatically places them into the reserved account `HPSPOOL`. If you are the user who created the spool files, you may view and access them as if they were in your own group and account.

Three numeric codes designate the files as input spool files, output spool files, and checkpoint files. They are:

1515	input spool files
1516	output spool files
1517	checkpoint files

For more information, see the discussion of the `BUILD` command later in this chapter.

Input spool files

The spooler creates input spool files when you submit jobs or enter data either via command line input (i.e. issuing the JOB or DATA commands) or from a spooled input device. The spooler copies a streamed or input spooled job to an input spool file and MPE schedules the job. When the job logs on, the spool file becomes the job's input ($STDIN). See the JOB command in the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information. Similarly, data are placed into an input spool file and can be accessed by the *user.account* specified in the DATA command. See the DATA command in the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information.

The system creates input spool files in the IN group of the reserved account HPSPOOL. Input spool file names have the format I*nnnn*.IN.HPSPOOL where *nnnn* is a number, for example, I235.IN.HPSPOOL. Because input spool files are always linked to the spooling subsystem, IN.HPSPOOL is the only place in the system containing input spool files. These files remain in IN.HPSPOOL until a job or process uses them or you delete them with one of the following methods:

- ABORTJOB for a job's $STDIN spool file.
- SPOOLF with the DELETE parameter for input spool files created with the DATA command.
- A START NORECOVERY system startup deletes all JOB and DATA input spool files.

Note

Job input spool files have a one-to-one correspondence with job master table (JMAT) entries. The JMAT is rebuilt for updates and START NORECOVERY. Whenever an update or START NORECOVERY occurs, the system purges all input spool files. Input spool files are only *recovered* during a START RECOVERY startup because the JMAT is only recovered at that time.

Output spool files

Output spool files normally reside in the OUT group of the HPSPOOL account and, unless you explicitly delete them, remain there until they print. You may use the ;SPSAVE parameter with the FILE, JOB, and SPOOLF ... ;ALTER commands to leave your spool file in OUT.HPSPOOL after all copies of the file print. Then you may copy the saved spool file into your own group and account so that you do not have to run the generating application again. You may not use the ;SPSAVE parameter to save a *private* output spool file.

Output spool file names have the format O*nnnn*.OUT.HPSPOOL where *nnnn* is a number, for example, O46.OUT.HPSPOOL.

The OUT.HPSPOOL group contains only linked output spool files. If you copy an output spool file from OUT.HPSPOOL to your account, the copy is not linked into the spooling subsystem. If you issue the SPOOLF command with the parameters PRINT and DEV on the copy, another copy is made in OUT.HPSPOOL and *this* copy is linked.

You can also *create* unlinked output spool files by using the BUILD or FILE commands with the ;SPOOL parameter or with the HPFOPEN intrinsic.

Checkpoint files

Checkpoint files are companions to output spool files that help the spooler recover from device problems such as power failure and paper jams. Checkpoint files also help a suspended spooler resume producing output. There is one checkpoint file per output spool file for each device that prints the spool file.

The output spooler creates the checkpoint file at the time it *begins* to print an output spool file, not before. The checkpoint file is automatically deleted when you or the spooler delete the output spool file from the HPSPOOL account or after a spool file that is saved with the ;SPSAVE parameter has its last copy printed.

The naming convention used for a checkpoint file is either of the following:

C*nnnnn.device_name*.HPSPOOL

or

C*nnnnn*.D*mmmmmmmm*.HPSPOOL

Where C*nnnnn* is the numerical value of the spoolid of the corresponding spool file preceded by a "C", and D*mmmmmmmm* is the logical device number (with leading zeros as required), preceded by a "D". Refer to the "File naming convention" section in appendix D for more information on naming a checkpoint file.

Checkpoint files are created for every output spool file, but they are only used on CIPER protocol printers.

Private and nonprivate spool files

All input spool files are automatically created private. By default, an output spool file is nonprivate, but you may choose to make it private for greater data security.

Private spool files

Private output spool files differ from regular nonprivate spool files in the following ways:

- Since they are level 2 privileged files, you may access them only by processes that call the `HPFOPEN` intrinsic while running at level 2 privileged mode. This means that the MPE/iX commands `PURGE` and `PRINT` fail. Level 3 programs such as `FCOPY` or your favorite editor and the `FOPEN` intrinsic can not open a private file.
- You may not save a private file by using the `;SPSAVE` option with the `SPOOLF ... ;ALTER` command.
- Users with SM capability cannot store private spool files, but they can use the `PRINTSPF` utility to print the files, and, if necessary, alter the target device.
- You may not copy, browse, or open a private output spool file (or an input spool file) as a disk file.
- You may not alter the number of copies.
- The only other control that you have over a private output spool file is to alter its priority, to defer or not defer it, or to delete it completely. You must have access to the spool file—according to the guidelines for nonprivate spool files—for this control.

Nonprivate spool files

You may access nonprivate spool files according to the following guidelines:

- If you have SM or OP capability, or if you are logged on at the system console, you may access any nonprivate spool files. This means that you can read, delete, or alter a spool file using either the NMS commands and intrinsics or standard MPE/iX commands and intrinsics.
- If you have AM capability, you may similarly access any spool file whose creating user is in your account.
- If you are the creating user, you may access spool files that you create.

If you have read access to nonprivate spool files, you may store and restore them with the `STORE` and `RESTORE` commands, respectively. If you have write access, you may purge nonprivate spool files using `STORE` with the `;PURGE` option.

Creating spool files

There are many different ways to create spool files. This section provides a quick overview of some of them.

Using a text editor

To have the spooler create a spool file, you direct output to a device whose spool queue is open. For example, when you issue the command to send a file to the printer from within a text editing program, the spooler creates an output spool file containing, among other information, the data you want to print. Try this simple example to see:

1. Log on to your MPE/iX system and start the text editor of your choice
2. Type a few lines of text and save them as `MYFILE1`.
3. Issue the command to send the text to the printer. For example, if you are using EDIT 3000, you would enter the `LIST ALL, OFFLINE` command at the editor prompt.
4. Exit the editor and, at the CI prompt, type the `LISTSPF` command. You'll see information something like this:

```
SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#06620     S1183    LP         8      1 LP       READY        USER.ACCOUNT

INPUT SPOOL FILES            OUTPUT SPOOL FILES
ACTIVE   = 0;                CREATE   = 0;              READY    = 1;
OPEN     = 0;                DEFER    = 0;              SELECTED = 1;
READY    = 0;                DELPND   = 0;              SPSAVE   = 0;
                             PRINT    = 0;              XFER     = 0;
                             PROBLM   = 0;

TOTAL IN FILES   = 0;        TOTAL OUT FILES   = 1;
      IN SECTORS = 0;              OUT SECTORS = 16;

OUTFENCE = 6
```

Streaming a batch job

A second way to generate a spool file is to submit a batch job that includes, as one of the list of command it executes, a command for sending a report to the printer. You use the STREAM command, followed by the name of the job file, to submit a job. For example:

```
STREAM jobfile
```

Try these steps to create and stream a simple job that sends a few lines of text output to a printer.

1. Log on to your MPE/iX system and start the text editor of your choice
2. Type a few lines of text and save them as MYFILE1.
3. Using the text editor, create a job file called MYJOB1 by entering the information shown below. For the italicized words, make the appropriate substitutions. (For example, supply your user name in place of *user*.) If you don't have a user password, then enter the information in the form *user.account/accountpass*.

```
!JOB MYJOB1,user/userpass.account/accountpass;INPRI=9;RESTART;OUTCLASS=LP,1
!CONTINUE
!EDITOR
T MYFILE1
L ALL,OFFLINE
EXIT
!TELL user.account MYJOB1 IS DONE.
!EOJ
```

4. Save this file as MYJOB1 and exit the editor.
5. At the CI prompt, enter the STREAM command to submit the job and note the number your job is assigned. For example:

```
STREAM MYJOB1
```

6. To list your output spool file, enter the LISTSPF command. For additional examples of the LISTSPF output, refer to the "LISTSPF" section in chapter 5.
7. Stream the same job a second time and issue another LISTSPF command to see how the listing changes.

Using the FCOPY utility

A third way to create a spool file is to copy a file to a spooled printer with the FCOPY utility. For example:

```
FILE SPPRNT;DEV=LP;CCTL
FCOPY FROM=MYFILE;TO=*SPPRNT
```

Using the PRINT command

A fourth way to create a spool file is to use the PRINT command with a standard MPE/iX file. For example:

```
FILE SPPRNT;DEV=LP;CCTL
PRINT MYFILE;OUT=*SPPRNT
```

Using the BUILD command

To create an unlinked spool file, you use the SPOOL parameter of the BUILD command. For example, to create the spool file MYSPOOL, you would enter:

```
BUILD MYSPOOL;REC=-132,1,F,ASCII;DISC=3000,1,1;SPOOL
```

To enter data into this spool file from a file called MYFILE enter:

```
FILE MYSPOOL,OLD
PRINT MYFILE,*MYSPOOL
```

The ;CODE= parameter of the BUILD command accepts three file codes for spool files. The file code parameter sets a specified value in the file label, which determines the type of spool file that the BUILD command will create. Specifying one of these file code parameters *without* also specifying the ;SPOOL option does not make the file a spool file. You can enter one of these file codes in addition to specifying the SPOOL option.

The file codes you can enter and their corresponding *mnemonic* (in this case is a word-like combination of characters that is suggestive of and represents a file code) are listed below.

Integer	Mnemonic	Meaning
1515	INSP	input spool file
1516	OUTSP	output spool file
1517	CHKSP	output spool file checkpoint file

For example, to specify a output spool file code with the BUILD command, simply add the ;CODE= parameter as follows:

```
BUILD MYSPOOL;CODE=1516
```

Using the FILE command

You can use the SPOOL parameter of the FILE command to create an output spool file that is not linked to the spool file directory and, therefore, will not be printed. A spool file created in this way may be printed at a later date.

To use the ;SPOOL parameter simply add it to a file equation as follows:

```
FILE SPPRNT;SPOOL
```

The ;PRIVATE option of the FILE command also generates a spool file, but one that may be accessed in privileged mode only. Private spool files may not be saved or copied. They may only be purged, printed, or (within limits) altered using the SPOOLF command. The PURGE or COPY commands may not be used on private files. To create a private spool file, simply add ;PRIVATE onto a file equation for a *spool file*. For example:

```
FILE SPPRNT;PRIVATE
```

If the file is not already a spool file, then add both the ;SPOOL and ;PRIVATE parameters to make the file a private spool file, as follows:

```
FILE SPPRNT;SPOOL;PRIVATE
```

Using the JOB command

Another way to generate a private output spool file is to add ;PRIVATE to the JOB command. For example:

```
JOB MYJOB;PRIVATE
```

Saving a spool file

Once the printer finishes printing your file, the output spool file is deleted. You can direct the spooler to save the output spool file to the group and account OUT.HPSPOOL instead of deleting it by adding the SPSAVE option to the JOB command at the beginning of your job file. To make these changes, do the following:

1. Start the text editor of your choice and text in the job file MYJOB1.
2. Edit the first line of MYJOB1 to change "**;OUTCLASS=LP,1**" to "**;SPSAVE**". The line should now read:

   ```
   !JOB MYJOB1,user/userpass.account/accountpass;&
   INPRI=9;RESTART;SPSAVE
   ```

3. Save MYJOB2, exit the editor, and issue the STREAM command to submit the job.
4. When the job is complete, enter LISTSPF. Notice that the state for this job's listing is SPSAVE.

The *1* in the JOB option OUTCLASS=LP,1 specifies that the output spool file has an output priority of 1. This effectively delays the printing of the file until you raise the file's input priority to a value greater than the outfence. (The outfence determines the minimum priority a spool file must have to be eligible for printing.)

Copying a spool file

The spooler places input and output spool files in a special group and account reserved for that purpose, and not in your own group and account. Input spool files are stored in `IN.HPSPOOL` and output spool files are stored in `OUT.HPSPOOL`.

You cannot copy input spool files; they are reserved for the exclusive use of the spooler process. If you want to save an output spool file into your account, you may do so by copying this file from the group `OUT.HPSPOOL` with the `COPY` or `FCOPY` commands. Spool files copied in this way are unlinked from the spooling subsystem, although the original spool file in `OUT.HPSPOOL` remains linked.

For example, to copy the output spool file `O1121` from `OUT.HPSPOOL` to the `report` file in your account, you would enter:

```
COPY FROM=O1121.OUT.HPSPOOL;TO=report
```

You can use the `COPY` command to copy a spool file to a hierarchical file directory structure. You cannot use the `RENAME` command to move the file. Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information on copying files.

Changing a spool file's characteristics

You can change an output spool file's characteristics by using the `SPOOLF` command. For example, to change the priority of a spool file, do the following:

1. Issue the `LISTSPF` command to view your output spool files, and note the value listed in the `PRI` column.
2. Choose an output spool file from the list, note its number, and use the `SPOOLF` command to change the priority to 3:

   ```
   SPOOLF Onnnn;ALTER;PRI=3
   LISTSPF
   ```

   If you get an error message, make sure that you have entered the correct output spool file number *preceded by a capital O, and not a zero*.
3. To verify that your spool file's priority is now 3, issue the `LISTSPF` command.

In the next series of steps, you will use the same commands to increase the number of print copies and place the file in a deferred state.

1. Enter `LISTSPF` again and notice two columns of the display: `COPIES` and `STATE`.
2. Choose the spool file you want to change, note its number, and enter the following command:

   ```
   SPOOLF nnnn;ALTER;COPIES=3;DEFER
   ```

   If you get an error message, make sure that you have entered the correct output spool file number *preceded by a capital O, and not a zero*.

3. To verify the change, issue the `LISTSPF` command.

Deleting a spool file

You can use the `SPOOLF` command to delete one or more of your output spool files before they are printed. To do so:

1. Enter `LISTSPF` and choose an output spool file to delete.
2. Issue the `SPOOLF` command with the `DELETE` parameter to delete the file.

   `SPOOLF O`*nnnn*`;DELETE`

3. Issue the `LISTSPF` command once again to verify that the output spool file no longer appears on the list.

Viewing the output of a spool file

Input spool files are automatically labeled private and, ordinarily you may not view them. You can easily view output spool files using any of the following methods.

- using text editors except those with restricted file codes or record types
- using either of the CI `PRINT` or `FCOPY` commands
- using one the following utilities: `SPIFF` or `PRINTSPF` (which are part of the Fundamental Operating Software) or the `HPBROWSE` utility, which you must purchase separately.

The following sections explain more about each of these options for view output spool files.

Using a text editor

One way to look at the contents of your output spool file is to use a text editor such as EDIT/3000. If you use another editor, you must experiment with it to find out if it can display spool files. Some editors are unuseable because they have restricted file codes or record types.

For example, suppose that you created a text file using EDIT/3000 and sent it to the printer. If you then want to use EDIT/3000 to view the output spool file the spooler created, do the following:

1. Start EDIT/3000 and text in the output spool file. You may see this warning:

   `***WARNING*** RECORD SIZE TRUNCATED TO 255 BYTES`

   By default, EDIT/3000 displays only the first 72 characters in a record. Unless the largest record in your spool file exceeds 255 characters, you may ignore this warning. (To extend the line length visible in EDIT/3000, refer to chapter 4.)

2. To view the entire spool file, enter the `LIST ALL` command.

3. If you use EDITOR to modify the file, you *cannot* save the modified file with its original name in OUT.HPSPOOL. You may, however, save it in your logon group and account.
4. When you finish viewing the file, exit the editor.

Using the PRINT command

You can also view output spool files with the PRINT command. When you use PRINT, MPE/iX displays the spool file one screen at a time, without the overhead in each record.

If you have not copied the spool file to your local group and account, issue the PRINT command followed by the fully-qualified name of the spool file. For example:

```
PRINT O#nnnn.OUT.HPSPOOL
```

If you have copied the file to your local group and account, you may enter the command followed by the file name (without specifying your group and account), like this:

```
PRINT outspfile
```

Using the FCOPY command

You may also use FCOPY to display the contents of your output spool file on the screen. To do so, specify the name of the spool file in the FROM= parameter, and enter the TO= parameter without specifying a destination. For example:

```
FCOPY FROM = spool file ;TO =
```

Using the SPIFF utility

SPIFF allows you to view only those output spool files to which you have access and which are linked to the spooling subsystem. You cannot view a spool file that you have moved to your local group and account using SPIFF. To use SPIFF to view a spool file, do the following:

1. Run the SPIFF utility by entering the command:

   ```
   RUN SPIFF.PUB.SYS
   ```

2. After SPIFF displays its introductory banner, enter the following two commands to view the file:

   ```
   TEXT #Onnnn
   LIST ALL
   ```

   For example, if you want to view the output spool file identified as O1234, enter:

   ```
   TEXT #O1234
   ```

   If you get an error message, make sure that you have entered the correct output spool file number *preceded by a capital O, and not a zero*. Chapter 5 has a complete description of the SPIFF utility and its commands.

Using the PRINTSPF utility

The PRINTSPF utility allows viewing of both input spool files and output spool files. The standard MPE/iX user may not view input spool files, however, since you must have system manager (SM) capability to do so.

The advantage of using PRINTSPF is that it displays the spool file in a formatted manner so that you can examine the contents of both the data and the special overhead in each record.

To display one of your output spool files, enter:

```
PRINTSPF Onnnn
```

For example, if you want to view the output spool file identified as 01234, enter:

```
PRINTSPF 01234
```

If you get an error message, make sure that you have entered the correct output spool file number *preceded by a capital O, and not a zero*. Refer to the "Viewing spool files using the PRINTSPF utility" in chapter 4 for more information.

Using the HPBROWSE Utility

HPBROWSE is an optional utility that employs softkeys to enter commands. It permits string searching and handles long lines by permitting you to scroll left or right to view the text. If you have purchased the HPBROWSE utility, you may use it to view spool files. To use HPBROWSE, enter:

```
HPBROWSE filename
```

For more information, refer to *HP Browse/XL User's Guide* (36384-90001).

2

Spooler and Spool File Management Tasks

This chapter shows you how to use various commands and utilities to accomplish spooler and spool file management tasks. It deals primarily with output spool files because input spool files are managed by the system and generally don't require user intervention. The topics in this chapter include:

- spooler management tasks including starting, stopping, suspending and resuming a spooler, opening and shutting spool queues, and controlling user access to a spooled device.
- controlling the printing of report headers and trailers.
- printing on special forms.
- displaying information about specific spool files, a group of spool files, and a selected subset of spool files.
- altering a spool file, which includes changing its output device, priority, the number of copies and its deferred status, and how to print and delete spool files.
- transferring spool files between native mode systems using the `STORE` and `RESTORE` commands, and between compatibility mode and native mode systems using the `SPFXFER` utility.
- a discussion of spool file recovery at system startup.

Managing the spooler

The user who has access to the system console, typically the system operator and occasionally the system manager, can control virtually every aspect of spooling with the `SPOOLER` command. This includes starting, stopping, suspending, and resuming any spooler process, and releasing a suspended spool file so that another can be printed instead. The `SPOOLER` command may be entered *only* at the console unless it has been allowed to other users via the `ALLOW` or `ASSOCIATE` commands.

To use the `SPOOLER` command, you must always specify device(s) whose spooler you want to direct, and at least one other parameter that tells MPE/iX what to do. The general form of the `SPOOLER` command is shown below. The remainder of this section explains the various tasks you can do with the `SPOOLER` command and provides examples for you. For a detailed explanation of the `SPOOLER` command and its parameters, see chapter 4.

```
SPOOLER [DEV=] { ldev     }
               { devclass }
               { devname  }

{ ;SHOW                                                                    }
{ ;OPENQ [;SHOW]                                                           }
{ ;SHUTQ [;SHOW]                                                           }
{ ;START [;OPENQ] [;SHOW]                                                  }
{        [;SHUTQ]                                                          }
{ ;STOP  [;FINISH] [;OPENQ] [;SHOW]                                        }
{        [;NOW   ] [;SHUTQ]                                                }
{                  [;FINISH] [;NOKEEP]                                     }
{                  [;NOW   ] [;KEEP  ]                                     }
{ ;SUSPEND         [;OFFSET=[+]page]                                       }
{                                [-]                                       }
{                  [;OPENQ] [;SHOW]                                        }
{                  [;SHUTQ]                                                }
{ ;RESUME  [;OFFSET=[+]page] [;OPENQ] [;SHOW]                              }
{                           [-]      [;SHUTQ]                              }
{ ;RELEASE [;OFFSET=[+]page] [;OPENQ] [;SHOW]                              }
{                           [-]      [;SHUTQ]                              }
```

The design of the native mode (NM) spooler prevents full backward compatibility with the compatibility mode (CM) spooler commands. For instance, `SUSPENDSPOOL` and `SPOOLER ... ;SUSPEND` without the `;FINISH` option causes the spooler process to retain ownership of the spool file that it is currently processing. Such differences will be noted as appropriate throughout this chapter, and they are further explained in Appendix B.

Starting a spooler

To create and activate a new spooler process on a spoolable device, enter:

```
SPOOLER DEV=6;START
```

Or, because `DEV=` is optional, you could also enter:

```
SPOOLER 6;START
```

Stopping a spooler

To stop a spooler process, enter:

```
SPOOLER 6;STOP
```

This command also closes the spooling queues for logical device 6. To stop the spooler process and to leave the spooling queues open for logical device 6, enter:

```
SPOOLER 6;OPENQ;STOP
```

Suspending a spooler

To suspend a spooler process, use the SPOOLER command with the SUSPEND parameter or use the SUSPENDSPOOL command. When you issue either of these commands, the spooler process retains ownership of the spool file that it is currently processing, but pauses the output. While the spooler is suspended, users may continue to create spool files, but no printing takes place. The printer(s) whose spooler process(es) have been suspended will remain idle until you command the spooler(s) to resume. When you issue the SPOOLER ... ;RESUME command (with no offset specified), the printer resumes printing exactly where it left off

For example, to suspend spooling on all printers belonging to device class LP, you would enter:

```
SPOOLER LP ;SUSPEND
```

There are many options to the SPOOLER;SUSPEND and the SUSPENDSPOOL commands that allow you to control the interruption of printing and determine the spooler's behavior when printing resumes. These options are explained next.

Finish the report first

To suspend spooling but allow a spool file that is currently being printed to finish printing, enter:

```
SPOOLER 6;SUSPEND;FINISH
```

Discontinue printing immediately

To immediately suspend spooling *without* allowing the currently printing spool file to finish, or to keep the file for reprinting at a later time, enter:

```
SPOOLER 6;SUSPEND;NOW
```

Because ;NOW is the default option, it may be omitted.

Discontinue printing and keep the spool file

You can suspend spooling but allow the spooled device to retain ownership of the file it is currently printing. This way, the spool file will resume printing (exactly where it left off or at another page that you specify) when the spooler resumes. To do so, you would enter:

```
SPOOLER 6;SUSPEND;NOW;KEEP
```

The ;KEEP parameter is the default and can be used only when ;NOW is specified or taken as the default. As a result, you could achieve the same result by entering:

```
SPOOLER 6;SUSPEND
```

Release the spool file

You can suspend spooling and direct the printer to *release* the currently printing spool file. If a spool file is released, a different spool file may begin printing when spooling is resumed. Or, the released spool file could be printed by another spooler process.

To release a spool file, you use the `NOKEEP` parameter. `;NOKEEP` may only be used when `;NOW` is specified or taken by default. For example:

```
SPOOLER 6;SUSPEND;NOW;NOKEEP
```

Specifying where to resume

When you are printing to `CIPER` protocol devices and HP 2680 and HP 2688 page printers, you can specify on what page to resume printing. For these devices, a page is one physical sheet. (Pages are *not* defined for other devices, and the results of using the `;OFFSET` parameter are unpredictable for them.) To instruct the spooler where to begin printing, you use the `;OFFSET` parameter. If you don't specify the `;OFFSET` parameter, printing resumes at the page where it stopped. This is the default.

`;OFFSET` is valid only in the following cases:

- when the spooler is actively printing a file and `;SUSPEND` is used
- when the spooler is releasing a file (with the `;RELEASE` option) that it retained during a previous suspend
- when the spooler is resuming printing of a file (with the `;RESUME` option) it retained during a previous suspend

For example, to suspend spooling and to position *backward* three pages from the page that is being printed, enter:

```
SPOOLER 6;SUSPEND;OFFSET=-3
```

To suspend spooling and to position *forward* five pages in the current spool file, enter:

```
SPOOLER 6;SUSPEND;OFFSET=+5
```

To suspend spooling and to position seven pages from the *beginning* of the spool file, you would enter the command below. In this case, the *absence* of a `+` or `-` sign indicates an absolute offset from the *beginning* of the spool file.

```
SPOOLER 6;SUSPEND;OFFSET=7
```

To be sure that a spool file begins printing at its beginning, enter:

```
SPOOLER 6;SUSPEND;OFFSET=1
```

Suspending a network printer spooler

When using network printers, avoid using the following commands to suspend the spooler in mid-file:

```
SPOOLER ... ; STOP
SPOOLER ... ; SUSPEND; OFFSET=anything
SPOOLER ... ; RESUME; OFFSET=anything
SPOOLER ... ; SUSPEND; NOKEEP
SPOOLER ... ; RELEASE
```

Many interfaces drop a network connection if the printer is ready to receive data but no data is being sent within a specific time period. The period is configurable at the printer or in the printer's TFTP file (specified in the bootptab entry), but many users simply use the factory default, which is 90 seconds.

The timer only runs when the printer is available but the host is not sending data, as is the case during a mid-file suspension. The timer does not run when the printer is unable to print, i.e., it has been taken offline, or places itself offline due to a paper out or toner low condition.

Resuming a spooler

Use the NMS command `SPOOLER` with the parameter `RESUME` or the command `RESUMESPOOL` to resume processing a spool file. Both `SPOOLER ... ;RESUME` and `RESUMESPOOL` begin printing where the printer left off, provided that the following conditions are met:

- You suspended the spooler with either `SUSPENDSPOOL` (without the `;FINISH` option) or `SPOOLER` *ldev* `;SUSPEND` (without the `;FINISH` option but with the `;KEEP` option and with no specified offset).
- You did not enter `SPOOLER` *ldev* `;RELEASE` while the spooler was suspended. (The `;RELEASE` parameter directs a suspended spooler to release a spool file that it is currently retaining.)
- You did not specify an offset as part of the `RESUME` command.

Now suppose that logical device 6 is owned by a spooler process. Enter:

```
SPOOLER 6 ;SUSPEND
```

Suppose that the spooler had just transmitted line 20 of page 10 and suppose that the conditions above hold. To continue spooling at line 21 of page 10 as if the suspension never took place, enter:

```
SPOOLER 6 ;RESUME
```

If you do not suspend this way, the spooler prints a trailer and prints a header when it resumes. If the spooler releases a file, any specified offset is honored by the next spooler which prints the file. If you do not specify an offset, the next spooler starts at the beginning of the page at which the previous output was suspended.

Note

If you use the following RESUMESPOOL commands to interrupt printing of a spool file on a printer that does not support Page Level Recovery (PLR), the spooler displays a warning on your $STDLIST indicating it is initiating a recovery sequence. (Refer to appendix C for details on PLR.)

```
RESUMESPOOL ... ; BEGINNING
RESUMESPOOL ... ; BACK any PAGES
RESUMESPOOL ... ; FORWARD any PAGES
```

Releasing a suspended spool file

If the ;SUSPEND parameter is used with the ;KEEP option or the ;KEEP option is taken by default (that is, neither ;KEEP nor ;NOKEEP is specified), the spooler process *retains* the spool file which was printing when the command was entered. This means that the currently printing spool file is the first spool file to print when spooling is resumed. You may use the ;RELEASE parameter to *release* a retained spool file. A released spool file is closed and the spool file prints based on its output priority relative to other spool files. The released spool file may also be printed by another spooler process.

To release a retained spool file, enter:

```
SPOOLER 6;RELEASE
```

You may use the ;OFFSET option to specify the spool file location where printing begins when the spool file finally begins to print. To release a spool file and to specify that it is to begin printing 10 pages back from the current page position, enter:

```
SPOOLER 6;RELEASE;OFFSET=-10
```

To release a spool file but ensure that it resumes printing at its beginning, enter:

```
SPOOLER 6;RELEASE;OFFSET=1
```

You may also position the spool file forward from the *current page* position or forward from the *beginning of the spool file*. For information on how to do this, see the previous section on suspending spool files. The ;OFFSET parameter works in exactly the same way with each of the ;SUSPEND and ;RELEASE parameters. When you use the ;RESUME parameter, ;OFFSET works in the same way except for the case of a spooler that did not retain its file. ;OFFSET is not valid with ;START or ;STOP.

Displaying spooler process status

The SPOOLER command ;SHOW parameter displays the status of the spooling process. To issue this command, enter:

```
SPOOLER 6;SHOW
```

If a device class is specified, status is displayed for all spoolable devices in the class.

You may use ;SHOW with any combination of other SPOOLER command parameters. For example:

```
SPOOLER 6;SUSPEND;OFFSET=1;OPENQ;SHOW
```

;SHOW produces a listing similar to:

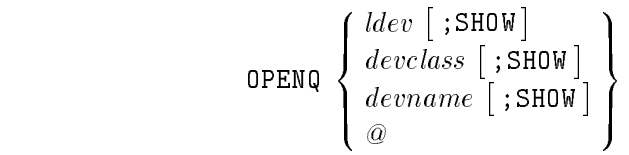
```
LDEV   DEV      SPSTATE     QSTATE   OWNERSHIP   SPOOLID
 6   0000006  *SUSPEND     OPENED   OUT SPOOLER  #0237
19   0000019  *ACTIVE      OPENED   OUT SPOOLER  #0264
```

The asterisk (*) indicates that the device is in a *pending* state on the way to the requested state. For example, suppose that you issue a SPOOLER command to *suspend* spooling. It is possible for the ;SHOW option of the command to finish processing before the spooler is fully suspended. In this case, an asterisk precedes the state of the spooler, shown under SPSTATE, to indicate that it is being changed.

Opening the spool queues

The OPENQ command opens the spooling queue for each device and makes it possible for users to create spool files. It does *not* enable you to print spool files. The OPENQ command may be entered only from the system console unless allowed to other users with the ALLOW or ASSOCIATE commands. Before spool files can print you must *start* the spooler.

The general form of the OPENQ command is:



For example, to open the spooling queue for all devices in class LP, enter:

```
OPENQ LP
```

To see information about the state of the queues and device(s) for which you are opening spooling queues, enter:

```
OPENQ LP;SHOW
```

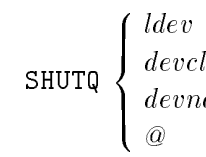
To open all spool queues that were shut because the system ran out of disk space, a file limit was encountered on the HPSPOOL account or its groups, or the SHUTQ @ command was entered, enter:

```
OPENQ @
```

Shutting the spool queues

The SHUTQ command prohibits anyone from creating new spool files but has no effect on spool files which have already been opened. It does *not* prevent spool files from printing. To do that, you must *stop* or *suspend* the spooler. The SHUTQ command may be entered only from the system console unless allowed to other users with the ALLOW or ASSOCIATE commands.

The general form of the SHUTQ command is:



For example, to close the spooling queue(s) for all devices in class LP, enter:

```
SHUTQ LP
```

To see information about the state of the queues and device(s) for which you are closing spooling queues, enter:

```
SHUTQ LP;SHOW
```

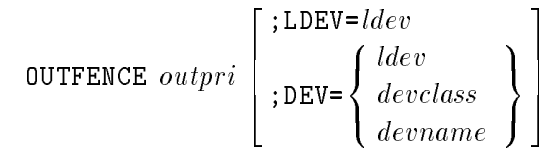
To shut all open spool queues on your system, enter:

```
SHUTQ @
```

Controlling printer access

The OUTFENCE command does not affect the spooler process directly but you may use it to control access to a spooled printer. The OUTFENCE command enables you to assign a *fence* or numerical barrier to one or more printers. An output spool file does not print unless its output priority exceeds the outfence. The OUTFENCE command may be entered only from the system console unless allowed to other users with the ALLOW or ASSOCIATE commands.

The general form of this command is:



The *outpri* parameter can be any value from 1 to 14. An outfence of 14 prevents any spool file from printing. The ;DEV parameter may be a logical device number, device class, or device name and is optional. The ;LDEV parameter refers only to a printer's logical device number. It also is optional. If ;LDEV and ;DEV are omitted, the command applies to all spoolable printers.

For example, to prevent anything from printing on all devices which are members of the device class LP, enter:

```
OUTFENCE 14;DEV=LP
```

To set the outfence to 8 for LDEV 6, enter:

```
OUTFENCE 8;DEV=6
```

Controlling the printing of headers and trailers

You may use the HEADON and HEADOFF commands to print or supress printing of a paper *header* and *trailer* page between each report. Paper headers may make it easier to physically separate and distribute printed material

To produce a printed header and trailer on a specific logical device, enter:

```
HEADON 6
```

To eliminate the header and trailer, enter:

```
HEADOFF 6
```

Either command takes effect when the next spool file is started.

Reprinting spool files

By default, spool files are deleted after the last copy is printed. The ;SPSAVE parameter causes a print file to be saved in the OUT group of the HPSPOOL account after the last copy of it has been printed. This is useful because it enables you to print spool files repeatedly (using the SPOOLF command) without having to rerun the producing application each time. If you do this, it may be desirable to *copy* the saved spool file to your own group and account to save space in the OUT group of the HPSPOOL account that is a shared system resource. You may then use the ;PRINT option of the SPOOLF command to print the file whenever you wish. To use this parameter simply add ;SPSAVE onto any file equation for printed output. For example:

```
FILE SPPRNT;SPSAVE
```

You may not use ;SPSAVE and ;PRIVATE together.

To save a job's $STDLIST output spool file in the OUT.HPSPOOL group after it has printed, add the ;SPSAVE parameter to the JOB command as follows:

```
JOB MYJOB;SPSAVE
```

Printing an unlinked spool file

The ;PRINT option of the SPOOLF command copies a linked or an unlinked output spool file to a *linked* output spool file that has an entry in the spool file directory. Once the file is *linked* to the spool file directory, it will be printed according to its *output priority*.

For example, to use the SPOOLF command to print the file MYSPOOL, you would enter:

```
:SPOOLF MYSPOOL;PRINT;DEV=LP
```

Printing on special forms

The ;FORMID parameter, in conjunction with the ;FORMS parameter, may be used to specify a unique special form for printed output. The ;FORMID parameter saves the operator from interacting with the console when multiple spool files are printed with the same special forms.

Unless you modify them with the FORMSALIGN command, the following rules apply to the ;FORMID and ;FORMS= parameters:

- If there is a specified FORMID different from the FORMID of the previous file that the spooler processed and if the FORMS=*formsmessage* is specified, then the forms message is displayed on the console or the $STDLIST of a user who has been *associated* to a spooled device with the ASSOCIATE command. The spooler waits for a reply to verify that the correct form is mounted and aligned before printing the spool file. The forms message is saved to use the next time there is no specified FORMID. The FORMID is also saved.
- If there is a specified FORMID different from the FORMID of the previous file that the spooler processed, if there is no FORMS=*formsmessage*, and if the spooler device has a nonstandard form from the previous file, then the standard forms message is displayed on the console or the $STDLIST of the associated user. The spooler waits for a reply to verify that *standard* forms have been mounted. The standard forms message is saved to use the next time there is no specified FORMID. The FORMID is also saved.
- If the FORMID is the same as the last time, no message appears on the console or on the $STDLIST of the associated user. The same form as the last time is used.
- If there is no FORMID specified and the FORMS=*formsmessage* is the same as the last time (this includes no FORMS=*formsmessage*), then no message appears on the console or on the $STDLIST of the associated user. The same form as last time is used.
- If no FORMID is specified and a different FORMS=*formsmessage* is specified, then the new message is displayed on the console or on the $STDLIST of the associated user. The forms message is saved. The spooler waits for a reply to verify that the correct nonstandard form is mounted.
- If neither FORMID nor FORMS=*formsmessage* is specified *and* there was a FORMS=*formsmessage* specified the last time (but no FORMID), then the standard forms message is displayed on the console or on the $STDLIST of the associated user. The spooler waits for a reply to verify that the standard form is mounted. The standard forms message is saved.

The form name specified with the ;FORMID parameter is used for printing and for management of spool files. You can group spool files having the same form name by using the ;FORMID keyword of the

;SELEQ parameter with the LISTSPF and SPOOLF commands, which are described later in this chapter.

The FORMS=*formsmessage* must end with a period or you get an error message. The FORMID must be no longer than eight characters and must begin with a letter. An example of FORMS= (notice the period) and FORMID follows:

```
FILE SPPRNT;FORMS=MOUNT TAX FORM.;FORMID=TAX1040
```

In this case, the FORMID is TAX1040. The LISTSPF command with the ;DETAIL parameter displays TAX1040 along with the spool file. Suppose that you are not the console user nor do you have SM, OP, or AM capability. To display the output spool files in your *user.account* that have the attribute TAX1040 and to display the name TAX1040 with the spool files, enter:

```
LISTSPF 0@;SELEQ=[FORMID=TAX1040];DETAIL
```

Other issues to be aware of include these:

- FORMIDs have no meaning for hot printers. No part of the hot printer path interprets FORMIDs, only the forms message associated with them.
- A specific FORMID is intended for a *unique* forms message. Use of more than one forms message with the same FORMID causes unpredictable behavior. For example, using one FORMID and two forms messages where one forms message is *no message* is not recommended. The spooler does not flag this as an error.
- You can use the ;SELEQ= parameter with the LISTSPF and SPOOLF commands to select spool files with no FORMID by specifying a null string. First set up a file equation for a file with no FORMID:

  ```
  FILE NOFORMID;DEV=LP,2
  ```

  After creating the output file, use ;SELEQ= in the following way:

  ```
  ;SELEQ=[FORMID=""]
  ```

  For example, to display the spool file with no FORMID, enter:

  ```
  LISTSPF;SELEQ=[FORMID=""]
  ```

  Because you have qualified the LISTSPF command to display only those spool files with no FORMID, the display on the screen is as follows:

```
SPOOLID   JOBNUM  FILEDES  PRI   COPIES  DEV    STATE   RSPFN    OWNER

#O1       S12345  NOFORMID   2        1  LP     READY            USER.ACCOUNT1
```

;SELEQ=[FORMID=""] works equally well with the SPOOLF command.

Viewing Data About Spool Files

Use the LISTSPF command to display information about input and output spool files. The set of spool files that you are allowed to see depends on your capabilities.

The general form of the LISTSPF command appears below. For a detailed explanation of this command and its parameters refer to chapter 4.

```
LISTSPF [ [IDNAME=] { spoolid                          } ]
        [           { (spoolid [ ,spoolid] ... )       } ]

[ [;SELEQ=] { select-eq      } ] [ ;DETAIL ]
[           { ^indirect_file } ] [ ;STATUS ]
```

Viewing specific spool files

Use the LISTSPF command with the IDNAME parameter, followed by the numeric spool file identifier, to display information about a specific output spool file. You can use this command if

- you created the spool file.
- you are the account manager (AM) of the creator's account.
- you have system manager (SM) or system operator (OP) capability.
- you are the console user.

For example, to view data about output spool file 357, you would enter:

```
LISTSPF IDNAME=0357
```

Or, you may omit the keyword IDNAME and the 0, and enter the command this way:

```
LISTSPF 357
```

Note that if you omit the 0, by default an *output* SPOOLID is assumed. To specify *input* SPOOLIDs, you must precede each identifier with I.

If there is more than one file you want information about, you can string together the SPOOLIDs on the command line like this:

```
LISTSPF IDNAME=357,375,458
```

Or, by grouping multiple ID names within parentheses you may omit the IDNAME keyword:

```
LISTSPF (357,375,458)
```

Viewing multiple spool files

Use the following command to display information about multiple spool files (those spool files residing in `IN.HPSPOOL` and `OUT.HPSPOOL`):

```
LISTSPF @
```

This command displays information for the following:

- All spool files in `IN.HPSPOOL` and `OUT.HPSPOOL`, if you are the console user, or if you have SM or OP capability.
- All spool files created by any user in your logon account, if you have AM capability.
- All spool files in your *user.account*, if you are a user other than a console user.

If you are the console user, to display information about *all output* spool files, you may also enter:

```
LISTSPF
```

If you are not the console user, you can use this same command to display information about all output spool files for your *user.account*.

Using wildcards to specify spool files

You may use wildcards in specifying the SPOOLID in the following way:

@	■ Specifies all spool files if you are the console user or have SM or OP capability. ■ Specifies input and output spool files by any user of your logon account if you have AM capability. ■ Specifies all spool files for your *user.account* if you are any other user.
O@	■ Specifies all *output* spool files if you are the console user or have SM or OP capability. ■ Specifies all *output* spool files created by any user of your logon account if you have AM capability. ■ Specifies all *output* spool files for your *user.account* if you are any other user.
I@	■ Specifies all *input* spool files if you are the console user or have SM or OP capability. ■ Specifies all *input* spool files created by any user of your logon account if you have AM capability. ■ Specifies all *input* spool files for your *user.account* if you are any other user.

@, O@ and I@ are mutually exclusive. In other words, you may use only one at a time.

If you are the console user or a user with SM or OP capability and you want to obtain information about all *output* spool files, enter:

```
:LISTSPF O@
```

If you are a user with AM capability, the `LISTSPF O@` command displays all *output* spool files created by users in your account. If you are any other user, the `LISTSPF O@` displays all the output spool files for your *user.account*.

If you are not the console user and do not have SM or OP capability, to obtain information about *input* spool files for your *user.account*, enter:

```
:LISTSPF I@
```

If you are the console user or you have SM or OP capability, `LISTSPF I@` displays *all* input spool files. If you have AM capability, `LISTSPF I@` displays all input spool files created by any user in your logon account.

Viewing spool file subsets

A useful feature of the `LISTSPF` command is the ability to define or *select* a subset of spool files for which to obtain information. This is accomplished by using a *selection equation* with the `;SELEQ=` parameter.

For example, suppose that you are the console user and you want to display spool file information for all output spool files with an output priority of less than 8. You would enter the selection equation shown below (including the brackets):

```
:LISTSPF;SELEQ=[PRI < 8]
```

If you have AM capability, this command displays information for all output spool files created by users in your logon account; otherwise, it displays information for all output spool files in your *user.account*.

Using AND and OR in the selection equation

You may use the *logical operators* AND and OR in the selection equation. If you use AND and OR in the same selection equation, AND takes precedence over OR unless you use parentheses to indicate otherwise. The files for which you see information depends upon whether or not you are using the console. If you are the console user, you will see information for all spool files that match the selection criteria. If you are not, you will see information for all output spool files created by your *user.account*.

For example, to display information for all output spool files that have an output priority less than 8 *and* that were sent to LDEV 6, you would enter:

```
:LISTSPF;SELEQ=[PRI < 8 AND DEV = 6]
```

Or, for example, to display information for all output spool files with priority less than or equal to 10 *or* with destination device EPOC you would enter:

```
:LISTSPF;SELEQ=[PRI <= 10 OR DEV = EPOC]
```

Excluding items in the selection equation

Use NOT to *exclude* specified items with the selection equation. You may use NOT with AND and OR.

For example, if you are the console user or if you have SM or OP capability, to select all spool files not created by MANAGER.SYS, enter:

```
:LISTSPF @;SELEQ=[NOT(OWNER=MANAGER.SYS)]
```

Or, to select output spool files that do not have a priority of 8, enter:

```
:LISTSPF;SELEQ=[NOT(PRI=8)]
```

Suppose that you are the console user or you have SM or OP capability. The following command line uses AND and selects all spool files created with formal file designator MRKTDATA that have not been routed to device class LP:

```
LISTSPF @;SELEQ=[FILEDES=MRKTDATA AND NOT (DEV=LP)]
```

Suppose that you have AM capability. The following command line uses OR and selects all spool files created by users in your logon account that have priority 8 or are not in the ready state:

```
LISTSPF @;SELEQ=[PRI=8 OR NOT(STATE=READY)]
```

Using an indirect file with a selection equation

The indirect file is a convenient way to avoid the extra keystrokes associated with often-used and complex selection equations. An indirect file is simply an ASCII file that contains the selection equation. When you use the LISTSPF command you enter the indirect file instead of the selection equation.

Suppose that you want to select all spool files sent to formal file designator MRKTDATA but not sent to device LP. You would put the selection equation shown below (including both left and right brackets ([ ]) into an ASCII file using any HP 3000 text processor. This ASCII file would then be your indirect file.

```
[FILEDES=MRKTDATA AND NOT (DEV=LP)]
```

In the following example, the indirect file is named INDFILE. To use the indirect file with the LISTSPF command, enter:

```
LISTSPF;SELEQ=^INDFILE
```

You may give your file any name that suits you. You must remember to always precede the indirect file with the ^ sign.

Using relational operators for the selection equation

You may use the following relational operators in the ;SELEQ equation:

=	equal
<>	not equal
>	greater than
>=	greater than or equal
<	less than
<=	less than or equal

Selection equation parameters

Use the relational operators with any of the following keyword parameters to construct the selection equation of your choice.

DEV *	LDEV number, device name, or device class name
FILEDES *	Formal file designator
SPOOLID *	Spool File identifier number
PAGES	Number of pages in spool file
FORMID *	Form name
STATE *	State
JOBNAME *	Job or session name
DISP *	Disposition (PURGE or SPSAVE)
COPIES	Number of copies requested
PRI	Output priority
JOBNUM *	Job or session number under which spool file was created
RECS	Number of records in spool file
OWNER *	Owner in *user.account* format
JOBABORT *	$STDLIST of job that aborted with no *continue* in effect (TRUE or FALSE)
DATE	Creation date in *mm/dd/yy* or *mm/dd/yyyy* format

Keywords marked with the asterisk (*) can only be used with two relational operators, = and <>.

Using wildcard characters in the selection equation

Wildcards are also supported in selection equations specifying *owners*. Use the @ sign to represent any combination of characters. For example, if you are the console user and you want to select all output spool files created by *any* user in the MFG account, you would enter:

```
LISTSPF;SELEQ=[OWNER=@.MFG]
```

If you want to do the same thing but you are a user with SM or OP capability and not a console user, enter:

```
LISTSPF @;SELEQ=[OWNER=@.MFG]
```

In a selection equation that specifies a job number, you may use J@ and S@ to specify all job numbers and all session numbers respectively. If you do not have SM, OP, or AM capability nor are you the console user, the following example displays all your spool files that were created by a job:

```
LISTSPF @;SELEQ=[JOBNUM=J@]
```

This command displays *all* spool files that were created by a job if you are the console user or if you have SM or OP capability. If you have AM capability, this command displays all spool files that were created by a job in your logon account.

You may also specify a specific job or session number.

Displaying summary data about spool files

Used with only the ;STATUS option of the LISTSPF command, the display is limited to a statistical summary of spool file data, known as a *status* display. If you are not the console user, to see *only* this summary for the spool files for your *user.account* without listing the spool files, enter:

```
:LISTSPF ;STATUS
```

If you are the console user, LISTSPF ;STATUS displays the status for *all* spool files. If you are not the console user, to see this summary for *all* spool files to which you have access without listing the spool files, enter:

```
:LISTSPF @ ;STATUS
```

You cannot use ;STATUS in combination with ;DETAIL.

Displaying detailed data about spool files

To display more *detailed* information regarding spool files, you may add the ;DETAIL parameter to the LISTSPF command as follows:

```
:LISTSPF IDNAME=0234;DETAIL
```

Spool file identification after a system reboot

Spool file job and session numbers are transposed from the J*nnn* or S*nnn* format to the J'*nnn* or S'*nnn* format whenever you perform a system `START` with the `NORECOVERY` option or whenever you import files to the system with `RESTORE` or the `SPFXFER` utility.

In the above command, `J@` or `S@` also selects output spool files with job/session identifiers in the J'*nnn* or S'*nnn* format. Specifying `J'@` or `S'@` selects only spool files with J'*nnn* or S'*nnn* identifiers. You may also select a single job or session using the J*nnn*, J'*nnn*, S*nnn*, or S'*nnn* format.

Viewing spool file data with LISTFILE

Another command that supplies information about files is the `LISTFILE` command. You may use this command on spool files just as you would on any other file. For example, to display all output spool files in `OUT.HPSPOOL` if you have access to them, enter:

```
LISTFILE 0@.OUT.HPSPOOL;FORMAT=5
```

You may use the name of a specific spool file instead of `0` followed by the wildcard (`@`).

The `LISTFILE` command displays MPE/iX file system characteristics such as the record size, block size, file code, security, creator, and access dates. The `LISTFILE` command also displays file information for those files named in the hierarchical file directories. Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information on hierarchical files and the `LISTFILE` command.

`FORMAT=5` is useful for spool files, both linked and unlinked, because it shows a spool file's target device. It is particularly useful for unlinked spool files since `LISTSPF` does not display information about them.

Managing Spool Files

The SPOOLF command lets you change the characteristics of spool files such as the device, the output priority, the number of copies to print, and whether or not the spool file should be saved or deferred. You may also use it to print or delete spool files.

Altering spool files

You may use the ;ALTER parameter of the SPOOLF command to alter the characteristics of spool files. The ;ALTER parameter may *not* be used concurrently with the ;PRINT or ;DELETE parameters described below. If none of ;ALTER, ;PRINT, or ;DELETE are specified, ;ALTER is taken as the default. The general form of the SPOOLF command with the ;ALTER parameter is as follows:

```
SPOOLF [ IDNAME= ] { spoolid                                  }
                   { (spoolid[ ,spoolid] ... )                }
[ [ ;ALTER ] [ ;SELEQ= { [ select-eq ]      } ] ]
                       { ^indirect_file     }
[ [ ;DEV= { ldev     } ]
          { devclass }
          { devname  }
[ ;PRI=outpri ] [ ;COPIES= numcopies ]
[ ;SPSAVE ] [ ;DEFER   ] [ ;SHOW ]
            [ ;UNDEFER ]
```

When you are use the ALTER parameter of the SPOOLF command, you have four choices for how to specify the spool files you want to change:

- specify one or more spool files by entering their spool file identification numbers (SPOOLIDs) on the command line. For example:

  ```
  SPOOLF IDNAME=357,375,458;ALTER;PRI=4
  ```

- use the *wildcard* symbol in SPOOLID to specify *all* output spool files if you are the console user or if you have SM or OP capability. For example:

  ```
  SPOOLF IDNAME=0@;COPIES=4
  ```

- write a selection equation that specifies which spool files to include or exclude in the operation. For example, this sample command would *select* output spool files with output priorities less than 8 and set them to 12.

  ```
  SPOOLF 0@;SELEQ=[PRI < 8];ALTER;PRI=12
  ```

- write a selection equation and put it in an indirect file, which you then specify on the command line. For example, if you created an indirect file named INDFILE, you would specify it on the SPOOLF command line like this:

  ```
  SPOOLF 0@;SELEQ=^INDFILE;ALTER;PRI=12
  ```

You may give your indirect file any name that suits you. Remember to precede the indirect file with the ^ sign in the SPOOLF command.

When you use selection equations, the files that qualify depend upon your capabilities and whether or not you issue the command from the console. The selection is made from *all output* spool files if you are the console user or if you have SM or OP capability. If you are an AM user, the selection is made from all output spool files in your logon account. If you are not the console, user nor have SM, OP, or AM capability, the selection is made from the output spool files in your *user.account*.

Selection equations are described in detail in previous sections of this chapter. Please refer there for more information.

Changing the output device

To alter the print device (to device class LP2) for three spool files, enter:

```
SPOOLF IDNAME=357,375,458;ALTER;DEV=LP2
```

Or, omitting the IDNAME=, enter:

```
SPOOLF (357,375,458);ALTER;DEV=LP2
```

You may specify a device class, as in the example, or you may specify a logical device number or device name.

Changing the output priority

To alter the output priority of all linked output spool files of which you are the owner (or, if you are logged onto the console, or have OP or SM capability, all output spool files on the system), enter:

```
SPOOLF 0@;ALTER;PRI=12
```

This command alters the priority of all output spool files in your logon account if you have AM capabilty.

Changing the number of copies

To alter the number of copies for one or more output spool files, enter:

```
SPOOLF (357,375,458);ALTER;COPIES=3
```

Saving a spool file

To save one or more output spool files, enter:

```
SPOOLF (357,375,458);ALTER;SPSAVE
```

or

```
SPOOLF (357,375,458);SPSAVE
```

The second example uses `;ALTER` as the default.

When an output spool file is *saved*, a copy of it remains in the `OUT` group of the `HPSPOOL` account after it is printed.

Deferring a spool file

To defer one or more spool files, enter:

```
SPOOLF (357,375,458);ALTER;DEFER
```

A *deferred* spool file does *not* print until it is *undeferred*.

When a spool file is deferred in this way, its priority is *not* changed. Instead, it is simply marked as deferred. (Its state is `DEFER`.)

Undeferring a spool file

To undefer one or more spool files, enter:

```
SPOOLF (357,375,458);ALTER;UNDEFER
```

An *undeferred* output spool file does print if its output priority exceeds the *outfence* of the printer to which it has been sent.

Performing multiple operations simultaneously

You may use any of the above keyword parameters simultaneously. You may save and undefer an output spool file and change its device, priority and number of copies all at once, as in the following example:

```
SPOOLF 327;ALTER;UNDEFER;SPSAVE;COPIES=4;DEV=6;PRI=11
```

Displaying results of the SPOOLF command

To see the results of your `SPOOLF` command, add the `;SHOW` parameter. For example:

```
SPOOLF 327;ALTER;UNDEFER;SPSAVE;COPIES=3;SHOW
```

The `;SHOW` parameter may be used with *any* combination of other `SPOOLF` parameters.

Printing spool files

You may use the ;PRINT option of the SPOOLF command to print output spool files. The ;PRINT option makes a linked copy of the specified spool file. Like the ;ALTER option described above, you may also use it to save; defer and undefer a spool file; and to specify the print device, the priority, and the number of copies. The general form of the ;PRINT option of the SPOOLF command is as follows:

```
SPOOLF [IDNAME=]{fileset (fileset[,fileset] ... )}
  [ [;PRINT][;DEV= { ldev devclass devname}]
  [;PRI=outpri]
  [;COPIES= numcopies]
  [;SPSAVE]
  [;DEFER ;UNDEFER]
  [;SHOW]                                      ]
```

All parameters for the ;PRINT option are used exactly as described above for the ;ALTER option (and produce the same results) except for the IDNAME. One or more *filesets* are required for the IDNAME. A file set has the general form:

```
filename[/lockword[.groupname[.acctname]]]
```

Wildcards are supported. Even so, all spool files in the file set must be output spool files. If a spool file is not an output spool file, the print option fails, and the command continues on the rest of the spool files. Files that are not spool files are ignored since SPOOLF applies only to spool files.

If the spool file name is not fully qualified (the group and account names are not given), the default is the user's current logon group and account. If any spool file has a lockword, it must be supplied with the command in batch mode; therefore, the spool file cannot be part of a set defined with wildcards. This restriction does not apply in interactive mode because the system prompts the user for each required lockword. If you do not supply the correct lockword, the print option on that spool file fails with a warning message, and the command continues to operate on remaining spool files.

Remember, each file you specify must be a valid spool file.

Printing a spool file that you create

Suppose that you create spool file 098 with the SPSAVE option. The file prints once, then remains in OUT.HPSPOOL instead of being deleted from the account because you instructed the spooler to save it after printing. To print another copy of the file, you would use the SPOOLF command, like this:

```
SPOOLF 098.OUT.HPSPOOL;PRINT;DEV=LP
```

The resulting linked spool file has the default priority of 8 and one copy is printed.

Printing a spool file in your logon group and account

Suppose that you have copied a spool file from the OUT group of the HPSPOOL account into the PUB group of the MFGRPTS account and named it MFGDATA. To print this spool file, while you are logged onto that group and account, you would enter:

```
SPOOLF MFGDATA;PRINT;DEV=LP
```

Printing a spool file from a different group and account

To print MFGDATA from a different logon group and account, add the spool file's group and account name as follows:

```
SPOOLF MFGDATA.PUB.MFGRPTS;PRINT;DEV=LP
```

You must have access to MFGDATA.PUB.MFGRPTS in order to print it.

Printing a spool file with a lockword

Suppose that the spool file MFGDATA contains a lockword. You could enter it as follows:

```
SPOOLF MFGDATA/LOCKWORD.PUB.MFGRPTS;PRINT;DEV=LP
```

Whenever the SPOOLF command is executed in *batch* mode the lockword *must* be supplied with the spool file name as in this example. In session mode the system prompts you for lockwords.

Other PRINT options

The ;PRINT option of the SPOOLF command offers all of the options described above for the ;ALTER option except for the selection equation (;SELEQ=). So you may specify the device, the output priority, and the number of copies to print as well as whether to save, defer, or undefer the spool file. A sample command follows:

```
SPOOLF MFGDATA;PRINT;DEV=6;PRI=9;COPIES=3;SPSAVE;DEFER;SHOW
```

These parameters are described above in the discussion of the ;ALTER parameter.

Using wildcards to print spool files

You may use *wildcards* in the file set. For example, if you have SM or OP capability, print copies of *all* spool files in the OUT group of the HPSPOOL account by entering:

```
SPOOLF @@.OUT.HPSPOOL;PRINT;DEV=6
```

If you do not have SM or OP capability and you enter this command, you get an error when the system encounters the first file in the file set to which you do not have access.

Deleting spool files

The ;DELETE option of the SPOOLF command allows you to delete *linked* spool files. The IDNAME and ;SELEQ parameters of the ;DELETE option of the SPOOLF command operate in precisely the same way as they do for the ;ALTER parameter previously described in this section. The general form of the SPOOLF command with the ;DELETE option is:

```
SPOOLF [IDNAME=] { spoolid                          }
                 { (spoolid[ ,spoolid] ... )        }
[ [ ;DELETE] [ ;SELEQ= { select-eq      } ] ]
[ [ ;SHOW  ]           { ^indirect_file } ]
```

You may use the *wildcard* symbol in IDNAME to specify *all* output spool files. For example:

```
SPOOLF IDNAME=O@;DELETE;SHOW
```

Similarly, you could use SPOOLF O@;DELETE to delete all spool files to which you have access. For example, if you have SM or OP capability, it deletes *all* output spool files on the system. *Because this form of the command is so powerful, be very judicious when using it.*

```
SPOOLF O@;DELETE;SHOW
```

Deleting one or more spool files

To delete one or more spool files, enter:

```
SPOOLF (357,375,458);DELETE;SHOW
```

If you are using SPOOLF to delete input data spool files, you must use the format I*nnn* for the spool file identification.

Also, $STDIN input spool files cannot be deleted with the SPOOLF command. These files can be deleted only by issuing an ABORTJOB command against the job number to which the $STDIN is associated. Under normal circumstances, you seldom need do this, because the system deletes $STDIN spool files when their associated job terminates.

Using a selection equation to delete spool files

You may use selection equations with the ;DELETE option to delete a subset of spool files. Here is an example:

```
SPOOLF O@;DELETE;SELEQ=[PRI<8]
```

You may put your selection equation in an *indirect* file. Indirect files are described earlier in this section under the SPOOLF ;ALTER parameter.

Transferring spool files between systems

You may transport NMS nonprivate output spool files between NMS MPE/iX systems with `STORE` and `RESTORE`. You need not be a system manager to use these programs to store and restore spool files, but you must have nonshareable device (ND) capability.

However, if you are working with spool files that were created on an early version of MPE/iX (or classic MPE/V) that does not support the Native Mode Spooler, you use a different method to transfer files. Read "Using the SPFXFER utility" later in this section for that information.

Storing spool files

Linked spool files reside in the HPSPOOL account, but users in other accounts create them. Your ability to store linked spool files depends upon your capabilities. For example, if you are the creating user, you can store your own spool files even though they reside in HPSPOOL and not your account. If you are an account manager, you can store any spool files created by a user of your account. If you have system manager (SM) or system operator (OP) capability, you can store all linked spool files.

When you store a file on tape using the `STORE` command with the `PURGE` option (`STORE ... ;PURGE`), the system purges the spool file after storing it. The system also deletes the spool file directory (SPFDIR) entry and deletes any checkpoint files associated with the spool file.

If a spooler process finishes its last copy of a spool file that is in the process of being stored on tape, the spooler cannot delete the spool file. The file management routines leave the file in the `DELPND` state, where it remains until any one of the following occurs:

- Someone opens and closes the spool file (for example, with the `PRINT` command to display the file on `$STDLIST`).
- You use `STORE` with the `PURGE` option.
- The spool file is put into a Ready state by raising the number of copies with the command `SPOOLF ... ;ALTER ;COPIES=` *number* where *number* exceeds the number already printed.

Restoring spool files

Stored linked and unlinked spool files can be restored to become linked or unlinked. Any spool file restored using `RESTORE` into `OUT.HPSPOOL` becomes linked. If a spool file is restored elsewhere, it becomes unlinked. To avoid potential ID and name conflicts, spool files restored to `OUT.HPSPOOL` (and, therefore, linked) are assigned new SPOOLIDs.

You can restore files to `OUT.HPSPOOL` in one of the following two ways:

- Restoring files stored to tape from the `OUT.HPSPOOL` group. If you have SM or OP capability, you can restore files created by any user even if that user does not exist on the system. If you have AM

capability, you can restore files created by any user in your account provided the user exits. If you are a general user, you can restore files that you created.

- Specifying the GROUP=OUT ;ACCOUNT=HPSPOOL option with RESTORE. You may be logged on anywhere, but you must have SM or OP capability. This method is not recommended, however, since it also restores files that are not spool files in the selected file set.

If you have SM or OP capability and you restore files to OUT.HPSPOOL, you must explicitly specify the CREATOR option with RESTORE to check that the creating user exists on your system. Otherwise, RESTORE restores the file even if the creating user and account do not exist. The specified creating user must also have nonshareable device (ND) capability whenever an SM or OP uses the ;CREATOR option.

If you have AM capability and you restore *linked* spool files created by other users in your account, those users must have ND capability at the time of the restore, or the restore fails for that file. You, the restoring user, also must have ND capability or you cannot allocate the tape drive to restore the files.

For spool files, the RESTORE options ;CREATE=GROUP and ;CREATE=ACCOUNT do not create the HPSPOOL account or any groups in it if they do not exist. You cannot restore any spool files to HPSPOOL if the account does not exist. Since linked spool files are associated with the HPSPOOL account and the creator's account, there is an ambiguity in ;CREATE=ACCOUNT. Since you should never purge HPSPOOL, RESTORE resolves the ambiguity by not restoring the spool file if HPSPOOL is missing.

Suppose OUT.HPSPOOL exists and you have SM or OP capability. If you specify CREATE, RESTORE creates the spool file creator's account and user if they do not exist. The file is restored to OUT.HPSPOOL.

If you specify ;CREATOR=*newuser* ;GROUP=OUT ;ACCOUNT=HPSPOOL, you change only the file's account not the creator's account. Suppose that USER.ACCT originally created spool file ABC and that ABC is stored from the ACCT account. Enter:

```
FILE TAPEFILE;DEV=TAPE
STORE ABC.USER.ACCT;*TAPEFILE
```

If you restore ABC using ;CREATOR=NEWUSER ;GROUP=OUT;ACCOUNT=HPSPOOL, the result is file SPOOLID.OUT.HPSPOOL and the creator is NEWUSER.ACCT.

Suppose that you restore ABC as described above. Enter:

```
FILE NEWTAPE;DEV=TAPE
RESTORE *NEWTAPE;ABC;CREATOR=NEWUSER;GROUP=OUT;ACCOUNT=HPSPOOL
```

Suppose 023 is the SPOOLID assigned to file ABC as it is restored. Then ABC is created on the system as 023.OUT.HPSPOOL and the creator is NEWUSER.ACCT.

If you restore a file to OUT.HPSPOOL and that file is destined for a device or class not configured on the target system, the file is put in the PROBLM state. The system links the file to the queue and creates the queue if necessary.

The ;SHOW=LONG option of the RESTORE command displays both the original SPOOLID and the new SPOOLID of spool files restored to OUT.HPSPOOL.

Refer to the *STORE and TurboSTORE/iX Manual* (30319-90001) for detailed information on storing and restoring files.

Using the SPFXFER utility

Spool files created on earlier versions of MPE/iX not containing NMS or on a classic HP 3000 (MPE V/E based operating system) have a somewhat different internal structure so they must be converted or *transported* before they can be used in native mode. Similarly, native mode spool files must be *transported* before being used in compatibility mode or on a classic HP 3000. The SPFXFER utility allows you to transport spool files back and forth between these different system types.

The SPFXFER utility reads tapes created only by itself or the SPOOK utility and writes tapes only in a format readable by SPOOK or itself. SPOOK is a contraction of the words "spooler look" and is a utility available on classic HP 3000's and on MPE/iX systems not containing NMS. In these two environments, SPOOK is the only method for transferring files to and from tape.

Transferring spool files to native mode

The INPUT command allows you to restore spool files that were transferred to tape using SPOOK onto your system. It also restores spool files previously stored with this utility. Restored spool files are placed into the OUT.HPSPOOL group and account as linked spool files and are assigned new SPOOLIDs.

To use the INPUT command, you must have SM or OP capability. Also, ND capability is required to access the tape drive.

The general form of the INPUT command is:

```
INPUT [ [ username[ .acctname] ] ]   ;*tapefile
      [ [ dfid [ , ... ] ]       ]
```

The INPUT command requires a tape device back reference. So before running the utility, set up a file equation for a tape such as:

```
FILE T;DEV=TAPE
```

To run the SPFXFER utility, enter:

```
SPFXFER
```

The prompt > appears. To see all commands available in SPFXFER, enter HELP at the prompt.

Input by user and account name. To input all spool files created under a specific user and account name, enter:

```
INPUT USER.ACCT;*T
```

To input all spool files created by all users in a specific account, enter:

```
INPUT @.ACCT;*T
```

To input all spool files created by a given user in your logon account, enter:

```
INPUT USER;*T
```

To input all spool files created by a given user name in *any* account, enter:

```
INPUT USER.@;*T
```

To input all spool files created by all users in all accounts, enter:

```
INPUT @.@;*T
```

The user name and account need not exist in the system directory nor does this command create them.

Input by DFID. The DFID is the identifier given to a spool file by MPE/iX systems not containing NMS and on a classic HP 3000.

To input a single spool file by DFID, enter:

```
INPUT #0357;*T
```

You may also string several DFIDs and you may omit the `#O` as follows:

```
INPUT 357,375,458;*T
```

If *username.acctname* and DFID are omitted, all spool files belonging to the logon user are input. For example:

```
INPUT ;*T
```

Transferring spool files out of native mode

The `OUTPUT` command enables you to store spool files from your native mode environment onto a tape in `SPOOK` format for use on MPE/iX systems not containing NMS or on a classic HP/3000.

To use the `OUTPUT` command, you must have SM or OP capability. Also ND capability is required to access the tape drive.

The general form of the `OUTPUT` command is:

`OUTPUT` [[*username* [. *acctname*]]] ;**tapefile* [;PURGE]
[*spoolid* [, ...]]

The `OUTPUT` command requires a backreference to a tape device. So before running the utility, set up a file equation for a tape such as:

```
FILE T;DEV=TAPE
```

To run the SPFXFER utility, enter:

```
SPFXFER
```

Since native mode spool files can be much larger than those spool files created on MPE/iX systems prior to version A.40.00 or spool files created on MPE V/E systems, you may not be able to move NMS spool files onto those systems.

Output by user and account name. To output all spool files for a specific user and then purge them, enter:

```
OUTPUT USER.ACCT;*T;PURGE
```

The ;PURGE parameter is optional and causes files to be purged from your system after being written to tape.

To output all spool files for all users in a specific account, enter:

```
OUTPUT @.ACCT;*T
```

To output all spool files created by a given user in your logon account, enter:

```
OUTPUT USER;*T
```

To output all spool files created by a given user in *any* account, enter:

```
OUTPUT USER.@;*T
```

To output all spool files created by all users in all accounts, enter:

```
OUTPUT @.@;*T
```

Outputting by SPOOLID. To output a single output spool file, enter:

```
OUTPUT 749822;*T
```

You also may output several spool files by stringing their SPOOLIDs. For example:

```
OUTPUT 749822,37721,482943;*T
```

You may add ;PURGE to purge the spool files from your system as they are written to tape.

If *username.acctname* and *spoolid* are omitted, all spool files belonging to the logon user are output. For example:

```
OUTPUT ;*T
```

Spool File Recovery At System Startup

The input SPool File DIRectory (SPFDIR) and the output SPFDIR, which are created by Progen near the end of the system startup (boot) process, are filled with information from spool files in IN.HPSPOOL and OUT.HPSPOOL. This provides a run-time "cache" for spool file management. The time necessary to create and fill these directories depends on the number of files in each HPSPOOL group and how the output spool files are distributed among various device queues (LP, CIPER, PP, and so on). Recovery will take the longest time when there are many output spool files in a single queue, for example, LP.

Input spool files usually consist entirely of job $STDIN files. The only other input spool file is the :DATA file, rarely used anymore. The number of input spool files is typically so small that the time spent recovering them to the input SPFDIR is not significant.

Recovery process improvements

There have been two improvements to the recovery process which greatly reduces the amount of time necessary to recover spool files:

- The boot process spends a small amount of time to assemble the list of spool files in OUT.HPSPOOL. In a test requiring the recovery of 9000 spool files, this time period was two minutes.
- All spool file recovery now takes place in a separate system process which continues until all output spool files have been recovered into the output SPFDIR (or discarded, if they cannot or should not be recovered), and then terminates. The boot process completes in parallel with the SPFDIR recovery process, and the system then becomes available to users.

Although spool file recovery at is a system process, it is created in the CS queue, which means that it competes with user processes once the system is available. However, as a system process, its priority does not decay, nor is it subject to being time-sliced. If the recovery process does not block itself periodically, user processes are starved. If it blocks itself too often, spool file recovery time is prolonged. To deal with this, the recovery process now pauses one second for every 200 SPFDIR entries it recovers. This forces the Dispatcher to allow some time for user processes to run, thus improving response. The tradeoff is that the recovery process takes longer to complete.

Note

Progen creates a file in the permanent domain, HPDISU00.PUB.SYS, for use by the recovery process. The recovery process purges this file before it terminates. If HPDISU00.PUB.SYS exists at system startup, Progen purges it to create the new one needed by the recovery process. Do not create a permanent file with this name.

Spooler behavior during recovery

There may be a period after the system is available to users when some spool files in OUT.HPSPOOL do not have an entry in the output SPFDIR. While some features of the spooling subsystem (described below) are affected during SPFDIR recovery, existing capability, resource limits, and security restrictions have not changed.

While the SPFDIR recovery process is running:

- Users can stream jobs without restrictions.
- Jobs can log on (that is, $STDLISTs can be created) without restrictions.
- Spooler processes can open, print, and delete spool files where an SPFDIR entry exists. Spool files in OUT.HPSPOOL whose SPFDIR entry have not yet been recovered cannot be selected for printing.
- All output spool file management commands (`ALTSPOOLFILE`, `DELETESPOOLFILE`, `LISTSPF`, `SHOWOUT`, and `SPOOLF`) are available, with the restrictions described under "Issuing spool file management commands," below.
- The `SPOOLF ... ; PRINT` command is not affected by the SPFDIR recovery process.

Once the recovery process has terminated, all existing features of the spooling subsystem are fully available.

Waking an idle spooler process

Recovery of an SPFDIR entry by the recovery process does not wake an idle spooler process even if the entry's priority is above the outfence. To deal with this situation, you may wait until a user creates a new spool file destined for the device managed by the idle spooler. When that spool file enters the READY state, the spooler is notified. It then prints all available files above the outfence. Or, you may wake the idle spooler process by issuing a command such as `SPOOLER 6; SUSPEND` followed by `SPOOLER 6; RESUME`. (To use this command, you must be at the system console or have been `ALLOW`ed the `SPOOLER` command, or have associated a class that includes LDEV 6.) Or, any user with access to a newly-recovered spool file can wake the spooler process for the device with `SPOOLF` command if the spool file's priority exceeds the system (or device) outfence.

Issuing spool file management commands

If you issue one of the spool file management commands for a a single spoolid whose SPFDIR entry is not yet recovered, it is treated the same as a non-existent spool file and you will see the following message:

```
Spoolfile "!" either does not exist on the system, or you have
insufficient capabilities to access it. (CIWARN 4563)
```

Note that a `LISTF` of this <Onnnn>.OUT.HPSPOOL displays the filename. The spool file does exist; only its SPFDIR entry does not, as yet.

If you issue one of the spool file management commands for a list of specific spoolids (for example, `LISTSPF #08072` or `LISTSPF (#08072, #07963, #08010)`) it searches for each individual spool file in the list. If it cannot find an SPFDIR entry for the file, it returns error -8039 (Cannot find the spool file).

If you issue one of the spool file management commands for a wildcarded fileset (such as `LISTSPF 0@`, or `SHOWOUT SP;JOB=@`), you will see information for only those SPFDIR entries that exist at the time the command is entered.

The `SPOOLF 0@; ALTER` and `SPOOLF 0@; DELETE` forms of the `SPOOLF` command are disallowed, and the following new message is displayed to any user attempting either of these commands:

```
'SPOOLF ;ALTER' or 'SPOOLF ;DELETE' of a wildcarded fileset is
disabled until the output spoolfile directory has been rebuilt
following a system startup. (CIWARN 4652)
```

When recovery is complete

When the output SPFDIR is fully recovered, the following message is displayed on the system console:

```
The system has finished rebuilding the
output spoolfile directory.
```

There is no change for OpenView console users. The above forms of the `SPOOLF` command are then re-enabled.

Managing the HPSPOOL Account

The native mode spooler's directory structure consists of the following:

Reserved Account:	`HPSPOOL`
Reserved Groups:	`OUT`
	`IN`
	All device name groups
Reserved User:	`MGR`

Controlling spool file disk allocation

The `HPSPOOL` account and all its reserved groups reside on the system volume set. The NMS creates them there. You should not relocate them to a private volume set.

You can, however, control on *which* of the system volumes spool files may be allocated disk space. When it determines where to allocate spool file disk space, the system looks for members of the system volume set that have been configured as volume class `SPOOL`. If at least one volume exists with volume class `SPOOL`, spool files are allocated disk space only on the one or more volumes configured as volume class `SPOOL`.

If none of the system volumes are in volume class SPOOL, spool files may be allocated disk space on *any* of the system volumes configured as DISC.

File space limits

Since spool files are normal MPE/iX disk files in an ordinary account structure, the configuration for NUMBER OF SECTORS PER SPOOL FILE EXTENT and MAX NUMBER OF SPOOL FILE KILOSECTORS does not apply and has been deleted from the SYSGEN utility. You may control the amount of disk space allocated to spool files by varying the HPSPOOL account file space limit. You may limit input and output spool file disk space usage independently by adjusting the IN and OUT group file space limit; otherwise, you may set unlimited file space limits on each group. The default file space limits set for the HPSPOOL account and its groups is unlimited file space.

Purging spool files from the IN and OUT groups

Normally, you will not have to perform any file cleanup for the IN and OUT groups of the HPSPOOL account, since spool files are automatically deleted once printed. However, if you must purge spool files, use SPOOLF O@;DELETE to clean out the appropriate group.

Never use PURGEACCT or PURGEGROUP to remove spool files from the HPSPOOL account or from the OUT and IN groups. You might disable the entire spooling subsystem. Spool File directory (SPFDIR) routines are used by high-level file access commands. Purging a spool file, for example, also deletes its spool file directory (SPFDIR) entry.

The PURGEGROUP and PURGEACCT commands access spool files at a lower level and do not use SPFDIR routines. These commands purge the spool files but leave orphaned SPFDIR entries. You may list these orphaned entries by using the LISTSPF command, but you cannot delete them with SPOOLF ... ;DELETE. The startup of the system deletes these orphaned entries as part of its recovery procedure.

Purging checkpoint files

When the spooler file management routines close a spool file following its final copy (whether the spool file is deleted or saved), all associated checkpoint files are deleted.

If you have sufficient capability, you may purge the checkpoint files with the PURGE command. If you should do this while the associated spool file is still linked to the spool file directory (SPFDIR), a spooler process printing the next copy of the spool file creates a new checkpoint file. This means that the spooler cannot use the file for rapid recovery, as it could have if you had not purged the first checkpoint file.

File security

File security for the HPSPOOL account and its groups are as follows:

HPSPOOL account:	(R,A,W,L,X:ANY)
Groups in HPSPOOL:	IN and OUT
	(R,A,W,L,X,S:ANY)
Device name groups:	(R,A,W,L,X,S:GU)

where R is read, A is append, W is write, L is lock, X is execute, S is save, ANY is any user, and GU is group user.

When the HPSPOOL account is created during system startup, a user called MGR for the HPSPOOL account is created. The existence of user MGR is required by the account creation process. MGR.HPSPOOL has only limited authority over spool files.

Access to users' spool files, including the ability to purge those spool files, is granted only to the creator of a spool file and to the manager of an account (AM) whose user creates the spool file in that account.

The user MGR and the HPSPOOL account should have passwords to prevent unauthorized access.

Caution

Never alter the account and group security provisions. They ensure the proper operation of the NMS commands and the other MPE/iX commands.

The OUT.HPSPOOL group

The NMS automatically creates the OUT.HPSPOOL group at system startup if the group does not exist already. OUT.HPSPOOL contains only *linked* output spool files. Other spool files may exist in other accounts but they are not linked because they do not reside in the HPSPOOL account nor do they have an entry in the spool file directory.

The IN.HPSPOOL group

The IN.HPSPOOL group contains *all* input spool files. Input spool files are always linked to the spooling subsystem; therefore, IN.HPSPOOL is the only place where you find them. The NMS automatically creates the group IN.HPSPOOL at system startup if the group does not already exist.

The device name groups

The device name groups contain all the checkpoint files for linked output spool files. Every output spooler creates its own device name group according to the following rules:

- If the device name begins with a letter, the group name is the same as the device name. For example, PP1 begins with a letter and, therefore, the group name is PP1.
- If the device name was not explicitly configured using SYSGEN, then the default device name consists of eight digits. Replace the first digit with a "D" and append the remaining seven digits. This,

then, is the group name. For example, the default device name for logical device 6 is 00000006. The device name group is `D0000006`.

Each spooler creates its device only if the group does not already exist. You must explicitly purge the group if you have sufficient capabilities and if the group is no longer useful (as when the spooling device has been removed from the system configuration).

The spooler process that owns the group creates and manages its checkpoint files. Each spooler process creates one checkpoint file for a specific output spool file no matter how many copies that process prints; therefore, if three different devices print copies of a spool file, then three checkpoint files exist, one in each device name group. If only one device prints three copies of a spool file, then only one checkpoint file exists.

When a spool file does not print completely for any reason (such as a device power failure, file deferment, device reassignment, spooler process suspension, or stopping), the next spooler process that prints the spool file on the same device uses the checkpoint file for rapid recovery. For devices supporting such recovery, output starts at the page after the last complete page printed before the interruption. Printing may start at another point if you enter the `OFFSET` option together with the `SPOOLER` command.

3

Configuring and Operating Network Printers

Before Release 5.5 of MPE/iX, the Native Mode Spooler (NMS) allowed many programs to share a single printer connected directly to the HP 3000. Starting with Release 5.5, the spooler now supports any Printer Command Language (PCL)-based printers attached to the HP 3000 via a TCP/IP network connection and a JetDirect interface card. A printer connected to the system in this way is called a "network printer." Users can access such printers *only* via the spooler, and not as "hot" or unspooled devices. (Application programs, on the other hand, can issue networking calls directly to such printers without going through the spooler.) Examples of the kinds of printers you can access via a network are:

- LaserJet series of laser printers, such as the LaserJet 4Si
- The inkjet series printers, such as the DeskJets and PaintJets
- The System Printer operation (SPO) HP5000/C30 and C40 cut sheet laser printers.

This chapter describes how to set up and operate network printers on the HP 3000. The first sections, intended for the system manager, list the supported devices and describe the two procedures needed to configure a network printer: using SYSGEN to add the device and creating the NPCONFIG.PUB.SYS configuration file. In addition, you will find some helpful configuration tips, two sample network printing configurations (one small, one large) and general information on spooler processes in a network printing environment.

The last part of the chapter is intended for anyone needing to operate a network printer. It explains how to access network printers, how to use special forms on network printers, and what text output is suitable for such printers.

Supported Devices

You must connect and prepare network printers according to the instructions furnished with your printer's hardware and with the printer's network interface. This information is not covered in this manual. However, the table below does list all of the HP devices that you can use in a network printing environment on an HP 3000. Some of the devices are listed by family, such as "PaintJet". Specific exceptions, such as LaserJet 4L, are listed separately; the family designation then applies to the rest of the family. The table also indicates whether or not Page Level Recovery and Page Count Logging are supported for each device.

Table 3-1. Support Networked Devices

Device/ Family	PLR	Page count
Color LaserJet	Yes	Actual
LaserJet 4 family (except 4L)	Yes	Actual
LaserJet 4L	No	Estimate
LaserJet III family	No	Estimate
LaserJet II family	No	Estimate
HP5000/C30	No	Estimate
HP5000/C40, without PJL support	No	Estimate
HP5000/C40, with PJL support	Yes	Actual
PaintJet, DeskJet, QuietJet, ThinkJet family	No	Estimate

Although the LaserJet IIISi supports a primitive level of PJL, the PJL is not sufficient to support Page Level Recovery or actual page count reporting.

Configuring a Network Printer with SYSGEN

The set of instructions below take you through the process of adding a network printer to your I/O configuration with SYSGEN. This is the first part of the network printer configuration process.

These instructions assume that you are an experienced system manager who has previously used SYSGEN. If you need more information, read *Performing System Management Tasks* (32650-90004). These instructions also assume that you have updated your system to a version of MPE/iX that supports network printing. (Network printer support is available on Release 5.5 and later of the MPE/iX operating system.)

Note

You may, if you wish, configure more network printers than you actually need at this time. To do so, read "Preconfiguring network printers" later in this chapter.

Adding a network printer to your configuration

To use SYSGEN to add a network printer to your system's I/O configuration, do the following:

1. Make sure you are logged on as MANAGER.SYS and run SYSGEN. At the CI prompt, enter:

   ```
   :run sysgen.pub.sys
   ```

2. At the sysgen prompt (`>`), enter `io` to start the I/O configurator.
3. Define the logical device identification for each network printer that you want to add. To use the default configuration values, specify `HPTCPJD` as the device identification and designate the path as `NONE`. For example, to configure LDEV 19 as a network printer, enter:

   ```
   io> ad ldev=19;id=HPTCPJD;path=NONE
   ```

   To view the device configuration, enter the `ld` command, for example:

```
io> ld 19

  LDEV:     19 DEVNAME:            OUTDEV:        0   MODE:          OS
    ID: HPTCPJD                    RSIZE:        66   DEVTYPE: PP
  PATH: NONE                       MPETYPE:      32   MPESUBTYPE:  0
 CLASS: NETLP
```

4. Enter the `hold` command to save the modified I/O configuration, and then type `exit` to leave the I/O configurator.
5. At the SYSGEN prompt, enter the `keep` command and then type `exit` to leave SYSGEN.

6. At a convenient time, perform an orderly shutdown of the system and then restart it to have the new I/O configuration take effect.

The following figure shows you a sample of the SYSGEN dialog needed to begin configuring LDEV 19 as a network printer.

```
:SYSGEN

SYSGEN version E.00.00 : catalog version E.00.00    WED, AUG 23, 1995,  4:16 PM
Copyright 1987 Hewlett-Packard Co. All Rights Reserved.

        ** First level command **

        io                log (lo)       misc (mi)        spu (sp)
        sysfile (sy)

        basegroup (ba)    keep(ke)       permyes (pe)     show (sh)
        tape (ta)

        clear (cl)(c)     exit (ex)(e)   help (he)(h)     oclose (oc)
        redo
sysgen> io

        ** IO configurator commands **

        aclass (ac)      adev (ad)       apath (ap)      avol (av)
        dclass (dc)      ddev (dd)       dpath (dp)      dvol (dv)
        lclass (lc)      ldev (ld)       lpath (lp)      lvol (lv)
        maddress(ma)     mclass (mc)     mdev (md)       mpath (mp)
        mvol (mv)

        clear (cl)(c)    exit (ex)(e)    help (he)(h)    hold (ho)
        oclose (oc)      redo
     io> ad ldev=19;id=HPTCPJD;path=NONE
     io> ld 19
 LDEV:     19 DEVNAME:                    OUTDEV:        0   MODE:          OS
   ID: HPTCPJD                            RSIZE:        66   DEVTYPE: PP
 PATH: NONE                               MPETYPE:      32   MPESUBTYPE:  0
CLASS: NETLP
     io> (...)
     io> hold
     io> exit
 sysgen> keep
 sysgen> exit

END OF PROGRAM
:
```

To complete the configuration of the network printer, you must add entries to the network printer configuration file NPCONFIG.PUB.SYS. Read "Creating the Network Printer Configuration File", later in this chapter, for information.

Preconfiguring network printers

If you have planned the expansion of your network printing capability, you can use SYSGEN to pre-configure the printers you will be physically adding at some future date. Other than counting toward the maximum number of devices on a system, there is no penalty for doing this. As you add the printers, you can write corresponding entries into the network printer configuration file (NPCONFIG.PUB.SYS) and place the printers in service without restarting the system.

To preconfigure network printers:

1. Use the I/O configurator in SYSGEN to define the additional network printers you will place in service at a later date. You may enter HPTCPJD as the device class name and accept the default values supplied for network printers. For example, to pre-configure LDEV 210 as a network printer enter:

   ```
   io> ad ldev=210;id=HPTCPJD;path=NONE
   ```

2. Leave the mode of the preconfigured printer set to output spooled (`MODE= OS`) and the path set to `NONE`. When the system boots, MPE/iX will try to create a spooler process for the printer. It will fail, however, when it cannot find an address for the device; see the next step. In this case, failure is appropriate since the printer does not exist.

3. When you are updating the NPCONFIG.PUB.SYS file to complete the configuration, do one of three things: (a) omit an LDEV entry for the preconfigured printer; (b) insert an LDEV entry, but make it a comment (i.e. "comment it out"); or (c) insert an LDEV entry for the preconfigured printer but do not include a network (IP) address for it. In this way, the spooler will not waste valuable CPU resources in a continual attempt to connect to a network printer that does not yet exist. Instead, the spooler process created for the printer will immediately terminate (and display a message) when it can't find the network address.

When you are ready to place the pre-configured printer in service, simply add the device-specific entry to NPCONFIG.PUB.SYS.

Creating the Network Printer Configuration File

The network printer configuration file NPCONFIG.PUB.SYS is a flat ASCII file that the system manager creates and modifies using a text editor. The purpose of the NPCONFIG file is to supply to the system additional configuration data about network printers that is not defined in SYSGEN. The NPCONFIG configuration file is designed to be extensible. As needed, for example, when placing a new network printer in service, the system manager may update the entries in NPCONFIG.

At a minimum, NPCONFIG must have the following information:

- An LDEV-specific entry for each network printer.
- The printer's network address, which is either its numeric IP address or a domain name which the spooler can resolve to its IP address.

For example, a complete though minimal entry in NPCONFIG for the network printer designated as LDEV 19 might be:

```
19      (network_address = 192.187.63.25)
```

Most NPCONFIG files will also have one global entry whose items are applicable to all network printers. This makes it convenient to configure a group of printers that belong to the same "family" since it is unnecessary to repeat identical configuration items for each LDEV entry. (With one exception, `setup_file`, items found in the LDEV specific entries take precedence over those in the global entry.) For example, a global entry for a group of LaserJet 4Si printers might look like this:

```
global (setup_file = LJ4SISET.HPENV.SYS      # LaserJet 4Si setup file.
        message_interval = 60                # Repeat msgs every >= 60 secs.
        banner_intray = 1                    # Upper tray has colored banner
                                             #   paper.
        banner_trailer = FALSE               # Only need a header page.
        pjl_supported = TRUE                 # LJ4Si is a full PJL device.
        jam_recovery = TRUE)                 # Reprints jammed pages by itself.
```

Any text that follows the pound sign (**#**) are comments, and can help make the NPCONFIG file self-documenting.

It isn't necessary to enter values for every possible item in either a global or LDEV-specific entry. Items that you do not specify automatically assume the default values. The "Items in an NPCONFIG entry" section, later in this chapter, describes each item in detail and includes its default value. For a brief overview of the items, refer to table 3-2.

For further reference, two sample configuration files and an explanation of the setup files they reference are included in this chapter. Read "A Small Sample Configuration" and "A Large Sample Configuration" for more information.

Syntax of NPCONFIG entries

When you add an entry to NPCONFIG, it must conform to a specific syntax. The entry consists of an `entry_id` which is either the keyword `global` for the one global entry in NPCONFIG, or the LDEV number of the network printer you are adding to the configuration (without leading zeros). The remainder of the entry is composed of keywords which indicate the item you are defining and the value you assign to it. So, the syntax of each entry is:

```
<entry_id> ( <keyword1>=<value1>
             <keyword2>=<value2>
             ...
           )
```

You may enter individual items in any order, one per line. Whitespace is optional except where required to delimit a token. All text between the pound sign (#) and the end of the line is treated as comment.

All text is case-insensitive except when it is part of a string. Thus, you can type the global entry specification as "GLOBAL", "global", "GlObAl", or any other combination of upper- and lowercase letters spelling "global".

For example, here is a global entry which sets the value of only one item, the poll interval:

```
global (poll_interval=5)
```

Here is an example of an entry for LDEV 19:

```
19     (poll_interval = 15
        network_address = 15.13.194.150
        program_file = OUTSPTJ.PUB.SYS)
```

Items in an NPCONFIG entry

The table on the next page briefly describes each of the items used in the NPCONFIG file and lists the default value. Following the table, each of the items is described in detail.

Table 3-2. Summary of NPCONFIG File Items

Item	Definition
`network_address`	Network address of the printer, specified either as an IP address or as a domain name that resolves to an IP address. This item is required. No default.
`TCP_port_number`	Port number used by the spooler to establish a TCP connection to the target printer. Default = 9100.
`program_file`	Name of the output spooler program file invoked for the network printer. Default is OUTSPTJ.PUB.SYS.
`poll_interval`	Initial time interval, in seconds, that the spooler waits to retry connecting to the printer. Default is 10 seconds.
`poll_interval_max`	Maximum amount of time, in seconds, that the spooler waits to retry connecting to the printer. Default is `poll_interval`.
`setup_file`	Fully-qualified MPE/iX name of a file containing information for a printer setup string. No default.
`run_priority`	Scheduling queue assigned to the output spooler process. Default is CS.
`SNMP_get_community_name`	The "SNMP Get Community Name" for this printer, which determines who can check printer status. Default is ALL.
`data_timeout`	Amount of time, in seconds, that the spooler waits for a specific network I/O request to complete before verifying that the printer and network are functioning correctly. Default is 10 seconds.
`snmp_timeout`	Amount of time, in seconds, that the spooler waits for a printer status check to complete. Default is 5 seconds.
`snmp_max_retries`	Number of times that the spooler should cycle through printer status checking before consulting `message_interval` to display error/warning messages. Default is 3.
`message_interval`	Minimum interval, in seconds, at which a printer status message is redisplayed, as long as it applies. Default is 0, which displays the message only once.
`banner_intray`	Sends a paper source command to the printer to allow banner pages to be taken from a separate paper tray. No default.
`data_intray`	Sends a paper source command to the printer which specifies the paper tray for normal data pages. No default.
`banner_header` and `banner_trailer`	Determines whether or not spool files are printed with a header, a trailer, both, or neither. Default is TRUE. The CI command `HEADOFF` overrides these settings.
`pjl_supported`	Specifies whether or not the spooler tests a printer to see if it can effect Page Level Recovery and actual Page Count Logging. No default.
`jam_recovery`	Specifies whether or not the spooler invokes its own jam recovery procedure. Default is FALSE.
`socket_trace`	When ON (enabled), initiates a socket-level trace of the TCP connection. Default is OFF.

Table 3-2. Summary of NPCONFIG File Items (continued)

Item	Definition
`transport_trace`	When ON (enabled), initiates a TCP-level trace of the socket in addition to a socket-level trace of the TCP connection. Default is OFF.
`default_page_size`	The Banner page size can be specified. Default is Letter.

Each of the items that you can use in an network printer entry in the NPCONFIG configuration file is described below.

Table 3-3. Description of NPCONFIG File Items

Item	Description
`network_address`	The network address of the printer, required for LDEV-specific entries but meaningless if used in a global entry. You may specify the address in one of two forms: ■ As an IP address consisting of four numeric fields in the form `aaa.bbb.ccc.ddd`. The address may be expressed in hexadecimal, octal, or decimal numbers. ■ As a domain name. If the `aaa` field begins with a letter, OUTSPTJ interprets *value* as a domain name, and tries to resolve it into an IP address. For more information, read "Entering a numeric IP address correctly," later in this chapter.
`TCP_port_number`	An integer between 1 and 32767 (inclusive) that identifies the port number by which the spooler establishes a TCP connection to the target printer. If you specify any other number (including 0) for `TCP_port_number`, OUTSPTJ will replace it with the default value of 9100. Optional. For example, to configure two printers on ports 2 and 3 of a JetDirect Ex+3 interface, configure the printer connected to port 2 with `TCP_port_number` 9101 and the printer connected to port 3 with `TCP_port_number` 9102.
`program_file`	The name of the output spooler program file invoked for the TCP/IP network printer, which is OUTSPTJ.PUB.SYS. The program file name is **case-sensitive** and must be fully-qualified. You may also specify this file name using POSIX syntax, i.e. /SYS/PUB/OUTSPTJ. If you enter any other name for `program_file`, you will see an error message at spooler process start time and no spooler process is created. (Refer to the "OUTSPTJ.PUB.SYS spooler program file" section for more information.) Optional.

Table 3-3. Description of NPCONFIG File Items (continued)

Item	Description
`poll_interval`	If the printer is unavailable to print a file, this is the initial time interval, in seconds, that the spooler waits before again attempting to connect to the printer. The default is 10 seconds. If you specify another value, it must be a positive integer. Optional. This value only has meaning when the spooler process is not suspended, when there is a file to print, and when the printer is busy or unavailable. This may happen, for example, if the printer's network interface is powered off or if another host on the network already has a connection to the printer. Do not confuse `poll_interval` with the polling performed by many third-party spooling solutions to determine if a spool file is available for printing.
`poll_interval_max`	The maximum amount of time, in seconds, that the spooler waits before again attempting to connect to the printer. The default is the current value for `poll_interval`. Optional. After the spooler fails to establish a network connection to its printer the first time, it waits the number of seconds determined by `poll_interval` before polling the printer again. Thereafter, each time it fails to establish the connection, it increases the poll interval by 25%, up to the absolute value specified by `poll_interval_max`. If the absolute value of `poll_interval_max` is less than `poll_interval`, the spooler uses the value of `poll_interval`. The spooler only uses the *absolute value* of `poll_interval_max` to limit the polling interval. The sign preceding the integer (`+` or `-`) determines whether or not a message that the limit has been reached is displayed on the console or Associated device. If you specify a negative value, the spooler displays a message. If you specify a positive value, no message is displayed.
`setup_file`	The fully-qualified MPE name of a file containing printer setup information. The setup file can be either a bytestream file or a record-oriented file. If the file is a record-oriented file, any carriage control (CCTL) byte is deleted, all leading and trailing blanks of each record are deleted and the remaining information is concatenated. The default is no setup file. If you insert `setup_file` in the global entry, it applies to all network print requests (unless the print request includes its own `ENV` specification). If `setup_file` appears in an LDEV-specific entry, the contents are *appended* to any specification in the global entry unless the print request includes its own `ENV` specification. The setup information can be anything that makes sense to the printer, including complex Printer Command Language (PCL) sequences or a combination of PCL and Printer Job Language (PJL) sequences for those printers that support PJL. *The spooler does not check the contents of the setup file.* Therefore, users should consult appropriate PCL and PJL documentation for their printer before attempting to construct a setup file.

Table 3-3. Description of NPCONFIG File Items (continued)

Item	Description
`run_priority`	The scheduling queue to which the output spooler process is assigned when it is created. Acceptable values are BS, CS, DS, or ES. The default value, CS, is used if you specify any value other than these choices. Spooler processes for system printers (HP2680, HP5000, etc.) are created in the BS queue because there are few such printers on a given system and because they require a constant stream of data to avoid warmup cycles and repositioning the paper. Spooler processes for serial printers (i.e. those connected via a DTC) always run in the CS queue and cannot be reassigned. Spooler process for network printers default to the CS queue, but you can assign them to another one. When running in the CS queue, spooler processes compete on an almost equal footing with user processes ("almost" because they are time-sliced, but as system processes they do not decay from the base of the CS queue as user processes do). However, if you have only a few serial or network printers, you may choose to run their spoolers in the BS queue. Hewlett-Packard generally recommends that you do not specify the DS or ES queues for network printers, since the resulting spooler performance would probably be unacceptable.
`SNMP_get_community_name`	The JetDirect interface card uses Simple Network Management Protocol (SNMP) services to retrieve printer status. SNMP allows you to restrict access to this retrieval service by assigning a password using the term "SNMP get community name." Most system managers allow the printer default "ALL" to remain in effect, which means that every user can access printer status. To ensure that the spooler can retrieve status information from the printer, set this item to the printer's "SNMP get community name". Note that if you set this item to anything besides `ALL`, you may wish to restrict read access to NPCONFIG.PUB.SYS to prevent users from reading the password. Optional.

Table 3-3. Description of NPCONFIG File Items (continued)

Item	Description
`data_timeout`	The amount of time, in seconds, that the native mode spooler waits for a specific network I/O request to complete before it checks whether or not the printer and network are operating properly. This item is one of four that you may use to determine how network and/or printer problems are managed. Default is 10 seconds, but you may enter another positive integer or 0. If you enter a value of 0 for `data_timeout`, the three other I/O timing items (explained below) are not used. Optional. The optimal value for `data_timeout` depends on the speed of the printer, the nature of the data (text, graphics, PostScript), the complexity of the network, and its normal bandwidth. For more information, read "Using I/O timing effectively" later in this chapter.
`snmp_timeout`	The amount of time, in seconds, that the native mode spooler waits for an SNMP request to complete before assuming that it will not do so and continuing. The default is 5 seconds. Optional. When the `data_timeout` expires, the spooler issues a Simple Network Management Protocol (SNMP) request to check the status of the printer. Since the SNMP request also must use the network, a network problem can also cause the expiration of `data_timeout`.
`snmp_max_retries`	The number of times that the spooler cycles through printer status checking before consulting `message_interval` (the next item) to possibly display error/warning messages. The default is 3, but you may enter a positive integer or 0. A value of 0 means that such messages are never displayed. Briefly, a single cycle works as follows: After transmitting data to the printer, the spooler waits for a period of time defined by `data_timeout`. If that timeout expires, the spooler issues an SNMP request to retrieve printer status, then waits a for a period of time defined by `snmp_timeout`. The item `snmp_max_retries` defines how many times the spooler executes this cycle before consulting `message_interval`. If the default value of 3 is used, for example, then the third `snmp_timeout` causes the spooler to consult `message_interval`.
`message_interval`	Specifies the minimum interval, in seconds, at which a printer status message (such as "LDEV #6, PAPER OUT") is displayed on the console or on an associated device, for as long as the message applies. Optional. The default `message_interval` is 0, which displays the message once, when the condition is detected, and not thereafter. You may enter a positive integer. For example, if you enter a value of 60 for this item, the `PAPER OUT` message will be redisplayed no more than every 60 seconds until the paper is loaded.

Table 3-3. Description of NPCONFIG File Items (continued)

Item	Description
`banner_intray`	Controls whether or not a PCL Paper Source command is sent to the printer, allowing banner pages to be taken from a separate paper tray. No default. Optional. The spooler performs no checks on the value you enter for `banner_intray` except to verify that it is an integer. **It is your responsibility to provide a valid positive value.** Typical values are 1 (selects the upper tray) or 4 (lower tray). Consult your printer's documentation for details. If you enter a positive integer, it is inserted into the PCL Paper Source command, Esc&*l*#H, replacing the "#", when a banner (header or trailer) is printed. If you omit this item, no Paper Source command is sent to the printer during banner printing, and any banner page is then taken from whichever intray was last used. If you enter a negative value or 0, the printer behaves as if the item were omitted. This allows an LDEV-specific entry to cancel a global entry. Once selected, the banner page intray is not automatically deselected to print data, nor can the spooler detect the tray that was in use prior to the Paper Source command that selected the banner tray. If you use the `banner_intray` item, you should also specify a value for the next item, `data_intray`. Alternatively, you can include a PCL Paper Source command that selects your normal data output tray in your `setup_file` or in a user-specificed ENV file. If the printer does not support multiple source tray selection, it ignores any Paper Source command sent to it. Therefore, it is safe to include this item in the configuration file for such a printer, especially if you intend to replace it soon with a printer with multiple source trays.
`data_intray`	When used in conjunction with the `banner_intray` item for a printer with multiple paper sources, the `data_intray` item instructs the printer to take banner pages and data pages from different intrays. No default. Optional. The spooler performs no checks on the value you enter for `data_intray` except to verify that it is an integer. **It is your responsibility to provide a valid positive value.** Typical values are 1 (selects the upper tray) or 4 (lower tray). Consult your printer's documentation for details. If you enter a positive integer, it is inserted into the PCL Paper Source command, Esc&*l*#H, replacing the "#", when a data page is printed. If you omit this item, no Paper Source command is sent to the printer before printing data, and the paper for data is then taken from whichever intray was last used. If you enter a negative value or 0, the printer behaves as if the item were omitted. This allows an LDEV-specific entry to cancel a global entry. If the printer does not support multiple source tray selection, it ignores any Paper Source command sent to it. Therefore, it is safe to include this item in the configuration file for such a printer, especially if you intend to replace it soon with a printer with multiple source trays.

Table 3-3. Description of NPCONFIG File Items (continued)

Item	Description
`banner_header` `banner_trailer`	Used to restrict banner printing to either a header or a trailer, or neither one. The default is `TRUE` for both items, which prints headers and trailers, but you may also set either or both to `FALSE`. The values are not case-sensitive. Optional. The `banner_header` and `banner_trailer` items are only tested if Headers have been enabled for the device with MPE's `HEADON` command. In other words, setting either item to `TRUE` does not override an MPE/iX `HEADOFF` condition. However, setting both items to `FALSE` renders an MPE/iX `HEADON` command meaningless.
`pjl_supported`	Specifies whether or not the printer supports the necessary Printer Job Language (PJL) features to effect Page Level Recovery (PLR) and actual (as opposed to estimated) Page Count Logging. Enter a value of `FALSE` to indicate that the printer does not support the necessary PJL to do correct PLR or actual Page Count Logging. Enter `TRUE` to indicate that the printer does support the necessary PJL commands. (The values are not case-sensitive.) To have the spooler automatically test the printer for PJL support, omit the item. No default. Optional. Be careful not to specify `TRUE` for a printer that does not support the required level of PJL. If you do, the spooler may never complete the print job (since it sends a PJL request and then waits for a corresponding reply, which it will never get from a non-PJL device). Or, if the spooler does complete the job, the output will be interspersed with PJL command text generated by the spooler. If you do not need Page Level Recovery or actual Page Count logging on a printer that supports it, you may wish to set `pjl_supported` to `FALSE` anyway. If you set this item to `TRUE` or if PJL support is discovered automatically, there may be a noticeable delay between printing the header and data, or between printing the data and the trailer. When you are configuring a LaserJet IIIsi, set the `pjl_supported` item to `FALSE` (instead of `TRUE` *or* instead of omitting the item altogether). The LaserJet IIIsi printer supports only a primitive level of PJL, not a full implementation.
`jam_recovery`	Specifies whether or not the printer has its own jam recovery procedure. The default, `FALSE`, indicates that the printer does not have its own recovery procedure and, when a jam occurs, will rely upon the spooler's. Optional. If your printer is configured with enough memory to reprint any pages lost due to a paper jam (with no help from the host), you should set this item to `TRUE`. Note, however, that if you specify `TRUE`, and the printer does not recover from a jam on its own, data may be lost. (This is why the more conservative value (`FALSE`) is the default—output may be duplicated but should never be lost.) If you adopt the default value or set `jam_recovery` to `FALSE`, you may get duplicate pages of output, especially if the printer does not support Page Level Recovery and has to restart the print job from the beginning of the file.

Table 3-3. Description of NPCONFIG File Items (continued)

Item	Description
`socket_trace`	Specifies whether or not socket-level tracing is on. The default is OFF. Optional. When this item is ON (enabled), it initiates a socket-level trace of the TCP connection. Socket-level tracing is useful when you suspect that the problem is in the bytestream of data sent to the printer as opposed to a problem that may be occuring at the TCP transport layer. (If you suspect that the problem is related to networking, using the `transport_trace` item, described next.) To format the data you receive from a socket trace, use the NMDUMP utility and, when prompted, specify subsystem number 5 (sockets).
`transport_trace`	Specifies whether or not TCP-level tracing is on. The default is OFF. Optional. When this item is ON (enabled), it initiates a TCP-level trace of the socket in addition to a socket-level trace of the TCP connection. Transport-level tracing is useful when you suspect problems with transport level software or, for example, if you need to identify TCP port numbers and/or connection parameters on either your local or the remote systems. To format the data you receive from a transport trace, use the NMDUMP utility and, when prompted, specify subsystem number 3 (NetTransport).
`default_page_size`	If this value is 26, then the A4 page size is selected for the Banner. Default is set to 2 which is Letter. default_page_size = 1 # for EXEC page size default_page_size = 2 # for LETTER page size default_page_size = 3 # for LEGAL page size default_page_size = 26 # for DIN A4 page size

Making changes to NPCONFIG

The spooler process for each configured network printer reads the NPCONFIG file once, briefly, when it first starts executing. This occurs when the system first boots if the LDEV is configured as `OS` or "output spooled" in SYSGEN. The spooler also reads the NPCONFIG file whenever a user starts a spooler process by issuing either a `SPOOLER` *nn*`;START` or a `STARTSPOOL` *nn* for the printer.

You may edit the NPCONFIG file at any time. The changes will take effect on a particular LDEV the next time that the spooler process starts for that printer. The changes will have no effect on a spooler process that is already executing.

To change the network printer configuration for a spooler process that is currently running, follow these steps:

1. Open the file NPCONFIG.PUB.SYS in the editor of your choice and modify the entry as needed.
2. Save your changes to the file and exit the text editor.
3. Issue the `STOPSPOOL` or `SPOOLER <ldev>; STOP` command for the LDEV whose configuration you need to modify.
4. Issue the `STARTSPOOL` or `SPOOLER <ldev>; START` command for the LDEV.

Errors in NPCONFIG

If you enter a value in NPCONFIG that does not meet the requirements for that particular option, OUTSPTJ will replace the erroneous value with the default, and display a message about the replacement.

For example, suppose you enter a number that is not within the proper range for *poll_interval*, which is 1 to 2147483647. When this error is encountered, the output spooler substitutes the default value of 10 and displays a message similar to this:

```
Output spooler, LDEV #19:  Check NPCONFIG.  The valid range of
item "poll_interval" is 1 to 2147483647.
The spooler will use the default value, 10.
Native Mode Spooler message 9041
```

Or, for example, if you enter a value other than `TRUE` or `FALSE` for an NPCONFIG item that requires a binary setting, the output spooler substitutes the default value and responds with another kind of message. For example:

```
Output spooler, LDEV #19:  Check NPCONFIG.  Valid values of
item "pjl_supported" are TRUE and FALSE.
The spooler will use the default value, FALSE.
Native Mode Spooler message 9042
```

Security and the configuration files

You create the NPCONFIG.PUB.SYS with the same security matrix as a typical file in the PUB group of the SYS account. As a result, only users with SM capability or a user logged on in PUB.SYS can make changes to the file or to its security matrix. If you plan to add an ACD to the file, do so **only** after careful consideration of its impact.

The security matrix for files in PUB.SYS are also appropriate for any setup files specified in NPCONFIG.PUB.SYS. This matrix allows any user to read the files, but prevents the user from changing or purging them. It isn't necessary, however, to store setup files in PUB.SYS. You can place them in any group, account, or hierarchical directory

since you must enter their fully-qualified file names in NPCONFIG anyway; for example, `setup_file = LJPORTRT.HPENV.SYS`.

Here is the standard security matrix for NPCONFIG.PUB.SYS, which is also appropriate for your setup files.

```
FILE: NPCONFIG.PUB.SYS

ACCOUNT ------  READ : ANY
               WRITE : AC
              APPEND : AC
                LOCK : ANY
             EXECUTE : ANY

GROUP --------  READ : ANY
               WRITE : GU
              APPEND : GU
                LOCK : ANY
             EXECUTE : ANY
                SAVE : GU

FILE ---------  READ : ANY        FCODE: 0
               WRITE : ANY      **SECURITY IS ON
              APPEND : ANY        NO ACDS
                LOCK : ANY
             EXECUTE : ANY
```

Creating and Using Setup Files

Any user, regardless of their assigned capability, may create files containing customized setup strings (analogous to environment files) to specify the printer operating mode for network printers. Such files can be used in one of two ways: If you have system manager (SM) capability and therefore can edit NPCONFIG, you can name such a file as the `setup_file` for a particular printer LDEV so that it becomes the default. For example, the following entry in NPCONFIG specifies the setup file LJ4SISET.HPENV.SYS for LDEV 19:

```
19      (network_address = 192.187.63.25
         setup_file = LJ4SISET.HPENV.SYS)
```

If you do not have SM capability, you can create a setup file and direct the spooler to use it while printing a specific spool file by entering the `ENV` parameter in the `FILE` command or the (HP)FOPEN intrinsic. When you do so, the spooler assumes that the setup file defines the entire printing environment; that is, your setup file supercedes any other setup file specified in the NPCONFIG.PUB.SYS. For example:

```
:FILE MYOUT;DEV=NETPRINT;ENV=PCLELITD
:FCOPY FROM=MYFILE;TO=*MYOUT
```

The MPE spooler expects the contents of setup files to be the raw data stream needed by the printer, either in bytestream or record-oriented format. Comments are not allowed. A bytestream file is sent "as is" to the printer. For record-oriented files, the spooler deletes any carriage control (CCTL) character, then trims leading and trailing blanks and concatenates records to arrive at the sequence sent to the printer.

If your printer supports PJL, the resulting data stream can include both PCL and PJL sequences, but they should be ordered so as to make sense to the printer.

Caution

Users are entirely responsible for the contents of setup files and their resulting effect on operation. The spooler does not interpret or alter them in any way (except to remove CCTL and blanks as described in this section). In particular, you should be familiar with PCL and PJL concepts as well as the features of your target printer before attempting to create a setup file.

The MPE/iX default environment

For backward compatibility, the MPE system default environment is landscape mode, courier font, 132 characters per line, 6 lines per inch, 60 printed lines per page with a three line top and bottom margin, and single-sided operation. This is often not appropriate for output printed on a cut sheet device such as the LaserJet 4Si. The following approach has been chosen to provide maximum flexibility, while not requiring individual attention for all non-default printing needs.

Setup strings

A "setup", or "setup string", is that sequence of (raw) PCL and/or PJL commands used to place the target printer in a specific operating mode. For example, the PCL sequence Esc&*l*0O selects portrait mode, while Esc&*l*1S selects duplex (long edge binding) operation.

You can include any valid PCL/PJL sequence, of any length, in your setup. It is sent just before the user data portion of the spool file and will be used to print the body of the report. Banner pages are printed using the default backward-compatible MPE/iX environment described earlier.

You enter setup strings into a text file with your editor of choice. If the resulting file is a bytestream file, it is incorporated without modification. If the file is a record-oriented file (fixed, variable, or undefined-length records), the spooler first removes any leading carriage control (CCTL) code in each record, then removes any leading or trailing blanks in the record, then concatenates whatever is left to form the final setup string.

Setup file hierarchy

There are four hierarchical levels of setup available to network printer users:

1. An ENV file specification issued via the `ENV=<filename>` keyword in a FILE equation or in an (HP)FOPEN intrinsic. If you use the `ENV` statement, it is applied at the time the file is opened and **it supersedes all other setup file specifications** described below.
2. A global setup file specification in NPCONFIG.PUB.SYS, which applies to all network printers and is applied at print time.
3. An LDEV-specific setup file in NPCONFIG.PUB.SYS, which is applied at print time. Its contents are appended to any global setup file specification, adding to or overriding the global print setup information.
4. The default MPE/iX environment, which you cannot modify.

If the spooler uses one or both of the setup files specified in NPCONFIG, the setup file(s) are attached when the file is printed, *not* when it is opened for printing. This means that if you preview the spool file using the SPIFF utility or other browser, you do not see default setup information. Also, any changes to either setup are reflected the next time the spool file is printed. The attributes are not bound to the spool file when it is created. When you direct output to a class such as LP, that designation is resolved to a specific LDEV (that is a member of that class) by the time the file is printed. It is that LDEV's setup file that the spooler uses.

Attaching setup files at print time also allows you to direct old archived spool files or spool files originally targeted for a non-network printer to a network printer. Such files simply inherit the current printing environment of the network printer.

If you specify a global or LDEV-specific setup file that does not exist, cannot be opened for read access, or returns an error while being accessed, the spooler posts a warning message to the console or Associated user and defers the spool file. If neither the global entry nor the LDEV-specific entry of NPCONFIG designates a setup file, the MPE system default environment is used.

Network Printing Configuration Tips

This section gives the system manager some tips for creating a valid network configuration file that will work well in your environment. The topics in this section include:

- Entering IP addresses correctly
- Setting appropriate poll intervals
- Using I/O timing effectively

Entering a numeric IP address correctly

For each network printer you are configuring, you must enter in the NPCONFIG file the printer's TCP/IP network domain name or its IP address in the form `aaa.bbb.ccc.ddd`. If you enter the IP address, each field must have a value that is less than or equal to decimal 255. The numeric base of each field in the IP address is then determined individually as follows:

- If the field begins with `"0x"`, the remainder of the field is (case-insensitive) hexadecimal.
- If the field begins with a leading zero followed by 0 through 7 inclusive, the remainder of the field is octal. A field beginning with `"0<anything else>"` is an error.
- If the field begins with `[1-9]`, the entire field is treated as decimal. To specify a decimal IP address, do not use leading zeros.

Listed below is one valid IP address expressed in many possible forms:

`10.13.194.150`	*all decimal*
`012.015.0302.0226`	*all octal*
`0xA.0xd.0xC2.0x96`	*all (case-insensitive) hex*
`10.015.0xc2.150`	*mixed*

These are invalid IP address specifications:

`10.13`	*Only two fields, four required*
`abc.def.ghi.jkl`	*Non-numeric, may be a domain name*
`10.018.194.150`	*"8" is not an octal digit*
`10.0b.194.150`	*"0b" is also an invalid octal spec*
`10.0xUsoft.194.150`	*"Usoft" is not valid hex*
`10.0xDEC.194.150`	*Valid hex, but > 255 decimal*
`10.def.ghi.jkl`	*Can't mix IP address, domain name*

Setting appropriate poll intervals

At a given time, only one host can be connected to a network printer and others must wait until the printer is available. If the spooler has a spool file to print, and either the network or the printer are unavailable, it may need to attempt connecting to the printer many times before it finally succeeds. You may configure the length of time between retries by defining values for the `poll_interval` and `poll_interval_max` items.

You must decide whether the default value for `poll_interval` (10 seconds) gives the host an unfair advantage or disadvantage over other competing host systems and, if so, specify a more appropriate value. For example, if `poll_interval` is long compared with other hosts, then those hosts stand a better chance of attaching the printer than this host. If the interval is comparatively short, other hosts may not be able to gain access to a network printer. However, if the interval is too short, this host may consume excessive CPU time doing the polling.

The spooler uses the combination of values set by `poll_interval` and `poll_interval_max` to determine the polling cycle on an unavailable network printer. For example, if `poll_interval` = 10 while the absolute value of `poll_interval_max` is 60, a sequence of unsuccessful connection attempts would wait 10, 12.5, 15.6, 19.5, ... 59.6, 60, 60, 60 ... seconds between each successive attempt. Once the connection has been made, the next failed connection starts at `poll_interval` once more.

If `poll_interval_max` is less than 0, a message is displayed to the console whenever this limit is reached. If it is greater than or equal to 0, no message is displayed. At first glance this behavior may seem counter-intuitive. If a network connection fails using a short interval, its chances of success appear to *decrease* as the interval increases. This is, in fact, the case, but there are situations for which this may be desirable. One example might be a network connection which must pass through many routers or other interfaces, any of which can fail. Or the network itself may be down. Repeatedly trying to establish a connection using a small `poll_interval` under these conditions wastes local resources as well as unnecessarily increasing network traffic. If situations like this do not arise in your environment, you may omit `poll_interval_max`. Successive connection attempts are then all separated by `poll_interval` seconds.

Using I/O timing effectively

Four I/O timing items that you may enter into the NPCONFIG file control the frequency of status checking for network printers. They are:

- `data_timeout`
- `snmp_timeout`
- `snmp_max_retries`
- `message_interval`

The default values for these items are suitable for text-based reports sent to a printer such as the LaserJet 4Si, which is separated from the HP 3000 by a small number of network devices (routers, bridges, etc.). If the printer is located at a remote site, or the data is more complex, you may want to enter a larger value to avoid excessive checking cycles. A low value (more frequent checking) causes any problem to be detected sooner, but substantially increases the use of CPU and network resources if normal I/O has not completed within that interval.

The remainder of this section describes two scenarios to suggest how you might set the I/O timing items to best manage some typical network printing problems.

First scenario

In this scenario, suppose that each I/O requires 15 seconds to complete. Assume that none of the I/O timing items are specified in NPCONFIG, which means that all default values are used. For reference, those values are:

`data_timeout` = 10 seconds
`snmp_timeout` = 5 seconds
`snmp_max_retries` = 3
`message_interval` = 0

Condition	Result
Output is printer bound	The `data_timeout`, which is set for 10 seconds, expires and the spooler issues an SNMP request to verify that the printer is on line. The SNMP request completes before `snmp_timeout` expires, and reports that the printer is online. The spooler therefore, restarts the `data_timeout` interval. The I/O completes five seconds later, 15 seconds from when it started. This scenario requires one SNMP request per data I/O request, and is therefore very wasteful of CPU and network bandwidth. For better efficiency, `data_timeout` should be reconfigured to a higher value, perhaps 20 seconds.

The printer is offline	The `data_timeout` timer expires and the ensuing SNMP request completes normally. No SNMP retries are required, so a "device offline" message is displayed. Because `message_interval` = 0, this message is not repeated, although the spooler continues to cycle through `data_timeout` and SNMP requests until the printer is placed online again. Suppose that after displaying the "device offline" message, a subsequent SNMP request times out three times. This is a different condition than "device offline", and so an appropriate message is displayed for it. Again, the second message is only displayed once because `message_interval` = 0. However, if subsequent SNMP requests complete normally, the "device offline" message is different from the SNMP timeout message, and so is displayed again, once.
The network is slow	Suppose that the first cycle completes: `data_timeout` expires and `snmp_timeout` expires. On the second cycle, `data_timeout` expires, but this time the SNMP request completes properly and verifies that the printer is on line. Since the spooler did not retry the required three times, the retry counter is reset and no message is displayed. If, instead, the network does not respond, the third `snmp_timeout` expires and an appropriate "network problem" message is displayed once. With the above item values, the message appears 45 seconds after the write operation to the printer (3 * (10 + 5)).

Second scenario

In this scenario, suppose that each I/O requires 15 seconds to complete, and that the I/O timing items have been set to the following values in NPCONFIG:

`data_timeout` = 20 seconds
`snmp_timeout` = 5 seconds
`snmp_max_retries` = 0
`message_interval` = 15

Condition	Result
Output is printer bound	Since the I/O completes within the required 20 seconds, no further checking is performed.
The printer is offline	The `data_timeout` expires. The ensuing SNMP request completes normally, reporting the offline condition. An appropriate message is displayed and another `data_timeout` cycle begins. When it expires, the SNMP request is repeated. Twenty seconds have elapsed since the last message, which exceeds `message_interval`, so the offline message is displayed again, and continues to be displayed every 20 seconds as long as the offline condition persists.
The network does not respond	The spooler cycles continuously between `data_timeout` and `snmp_timeout` but displays no message, because the network problem appears as a non-response to the SNMP request and `snmp_max_retries` is 0.

A Small Sample Configuration

This section shows you a sample network printer configuration that might be appropriate for a small network confined to one site with identical LaserJet 4Si printers. This example shows configurations for three such printers, LDEVs 6, 200, 210. One printer, LDEV 200, is used heavily for PostScript output, so it has been configured as two separate logical devices: LDEV 200, which uses a standard LaserJet setup file, and LDEV 201, which uses a PostScript setup file.

The sample NPCONFIG file

The NPCONFIG file for this small configuration is shown below. Comment text appears after the pound sign (**#**), and is useful for making the configuration file self-documenting.

```
#                              NPCONFIG.PUB.SYS
#  This configuration uses the default values for the following items.
#  They are not explicitly defined in this file.
#     program_file = OUTSPTJ.PUB.SYS
#     poll_interval = 10 seconds
#     poll_interval_max = poll_interval = 10 seconds
#     banner_header = TRUE
#     data_timeout = 10 seconds
#     snmp_timeout = 5 seconds
#     snmp_max_retries = 3
#
#  Since all the network printers are LaserJet 4Si, most of the remaining
#  configuration items appear in the following global entry.

global (setup_file = LJ4SISET.HPENV.SYS      # LaserJet 4Si setup file.
        message_interval = 60                # Repeat msgs every >= 60 secs.
        banner_intray = 1                    # Upper tray has colored banner
                                             #   paper.
        data_intray = 4                      # Lower tray has normal paper.
        banner_trailer = FALSE               # Only need a header page.
        pjl_supported = TRUE                 # LJ4Si is a full PJL device.
        jam_recovery = TRUE)                 # Reprints jammed pages by itself.
#
#  The next four entries define the configuration items appropriate to
#  each specific LDEV.

19     (network_address = 192.187.63.25)

200    (network_address = 192.187.63.121)    # Standard text personality.

201    (network_address = 192.187.63.121     # Same printer, different LDEV,
        setup_file = LJ4PSSET.HPENV.SYS)     #   PostScript setup file.

210    (network_address = 192.187.63.82)
```

The sample printer setup files

The setup files used in the small configuration example are explained below.

LJ4SISET.HPENV.SYS

The global setup file LJ4SISET.HPENV.SYS is a typical setup for portrait mode printing. It consists of several groups of settings which are first shown in tabular form in Table 3-4, then as they might actually appear in the setup file. This file completely defines the printer setup. You may choose to configure many of these settings at the printer itself and omit them from the setup file. If you do, they will be set as soon as the spooler sends the Esc E command sequence before printing begins.

The line termination attribute, Esc &k2G, needs a bit of explanation. Text files generated on UNIX systems typically include only the <LF> as a record separator. The <CR> must be supplied, otherwise each line of the output would stairstep down the page (and most likely be truncated at the right margin). The value shown for this attribute directs the printer to supply a <CR> whenever it receives an <LF> or <FF> (formfeed). Note that an actual <CR> in the data stream before the <LF> or <FF> is redundant and does not change the output.

Here is a typical record-oriented file containing the setup information. In this example, we have taken advantage of the PCL property of combining commands within the same parameter and group to reduce each such combination to one record of the setup file. The native mode spooler would trim leading and trailing blanks from each record and concatenate them to arrive at the final sequence actually sent to the printer.

```
Esc Z Esc E
Esc &l1x0u0z0o1s4h1l2a6d3e60F
Esc &k10h2G
Esc &a8l88M
Esc (8U Esc )8U
Esc (s0p12h10v0s0b4099T
Esc )s0p12h10v0s0b4099T
Esc &d@
```

For further details regarding PCL, consult your printer's PCL documentation.

Table 3-4. Setup elements for the LJ4SISET.HPENV.SYS Global File

Group	Attribute	Value	PCL Code
Printer Control	Display functions	Off	Esc Z
	Reset defaults		Esc E
Job Control	Number of copies	1	Esc &l1X
	Logical page origin on physical page	(x,y) = (0,0)	Esc &l0u0Z
Page Control	Orientation	Portrait	Esc &l0O
	One side/two sides	Duplex, long edge	Esc &l1S
	Paper source	Bottom tray	Esc &l4H
	Page size	Letter (8.5"x 11")	Esc &l2A
	Perforation skip	On	Esc &l1L
	Horizontal motion index (in units of 1/120 in. 10 = 12 chars/inch)	10	Esc &k10H
	Lines per inch	6	Esc &l6D
	Top margin	3 lines	Esc &l3E
	Left margin	8 columns	Esc &a8L
	Right margin	88th column	Esc &a88M
	Text length (defines bottom margin)	60 lines	Esc &l60F
	Line termination	CR=CR, LF=CR-LF, FF=CR-FF	Esc &k2G
Font specification (primary "(" and secondary ")" character sets)	Symbol set	ROMAN-8	Esc (8U Esc)8U
	Spacing	fixed	Esc (s0P Esc)s0P
	Pitch	12 cpi	Esc (s12H Esc)s12H
	Height	10 point	Esc (s10V Esc)s10V
	Style	Upright	Esc (s0S Esc)s0S
	Stroke weight	Medium	Esc (s0B Esc)s0B
	Typeface	Courier	Esc (s4099T Esc)s4099T
	Underlining mode	Off	Esc &d@

LJ4PSSET.HPENV.SYS

The contents of an LDEV-specific setup file are *appended* to those of any global setup file. This means that the setup data shown below, contained in the file LJ4PSSET.HPENV.SYS, is sent *after* the data shown above as LJ4SISET. This setup file merely switches the printer's language personality to PostScript so that it interprets the data properly.

The first line of the setup file is a Universal Exit Language command. The carriage return (#13, %15, $0d, chr(13)), and line feed (#10, %12, $0b, chr(10)) characters (shown as the CR and LF keycaps, respectively) in the PJL commands are significant and must be included as data. Do not rely on MPE or the printer to supply any of these characters as line terminators. The spooler concatenates only the separate data records in the setup file and sends the result to the printer as one unbroken data stream. It does not insert any data of its own. You may include annotations and comments in your files *only* if you precede them with the `COMMENT` PJL command, as shown in the example below. Do not use other symbols such as the pound sign (`#`), curly braces (`{}`) or slash-asterisk combinations (`&/*`) to indicate comments in a setup file. The spooler has no provision for ignoring them.

For this example, assume that all required PostScript, including any necessary non-resident font definition, is in the user's data stream.

```
Esc%-12345X@PJL CR LF
@PJL COMMENT Beginning PostScript Job CR LF
@PJL ENTER LANGUAGE = POSTSCRIPT CR LF
```

mode at the end of each print job. There is no need for a user-supplied "exit" setup file, nor is there any provision for the spooler to use such a file.

A Large Sample Configuration

The next example of a network printer configuration file might be appropriate for a large network with many printers at different sites. This example includes five such printers (listed below). Four are LaserJet 4Si printers and the fifth is a DeskJet 1200C used for making color prints and slides. "Local" and "remote" are used with respect to the site of the host HP3000. The network printers are:

- A local printer for MPE-style reports (landscape, 132 chars, 60 lines per page).
- Another local printer for raw PCL documents.
- The DeskJet 1200C, a local printer.
- A remote printer for portrait mode text-based reports and raw PCL documents.
- Another remote printer for PostScript documents.

The sample NPCONFIG file

The NPCONFIG file for this large network printing installation is shown below. Comment text appears after the pound sign (**#**), and is useful for making the configuration file self-documenting.

```
#                              NPCONFIG.PUB.SYS
#  This configuration uses the default values for the following items.
#  They are not explicitly defined in this file.
#
#     program_file = OUTSPTJ.PUB.SYS
#     banner_header = TRUE
#     poll_interval = 10 seconds
#
#  Since most printers are LaserJet 4Si, their major properties are defined
#  in the global entry.  Items affected by the behavior of local versus
#  remote printing are specified in the individual LDEV entries.

global (setup_file = LJGLOBAL.HPENV.SYS      # Primary setup for LJ 4Si.
        banner_intray = 1                    # Upper tray has colored banner
                                             #   paper.
        data_intray = 4                      # Lower tray has normal paper.
        banner_trailer = FALSE               # Only need a header page.
        pjl_supported = TRUE                 # LJ4Si is a full PJL device.
        jam_recovery = TRUE)                 # Reprints jammed paper by itself.

#  For local printers, the default values of the I/O timing items are used:
#
#     poll_interval_max = poll_interval = 10 seconds
#     data_timeout = 10 seconds
#     snmp_timeout = 5 seconds
#     snmp_max_retries = 3
#
#  The I/O timing values for remote printers depend on their network
#  "distance", and are specified in their LDEV-specific entries.

200 (network_address = ptr4.mycorp.com       # Local printer "printer 4".
     setup_file = LJMPE.HPENV.SYS            # MPE-style setup file.
     message_interval = 60)                  # Repeat msgs every >= 60 secs.

201 (network_address = ptr7.mycorp.com       # Local printer "printer 7".
                                             # (no setup file, prints raw PCL.
                                             #   Inherits global setup.)
     message_interval = 60)

202 (network_address = ptr8.mycorp.com       # The DeskJet 1200C printer.
                                             # (no setup, assume output varies
                                             #   enough that any setup is in
                                             #   output or ENV file.  Inherits
                                             #   global setup.)
     banner_intray = 0                       # No banner tray on this printer.
     pjl_supported = FALSE                   # Supports language switching only.
     jam_recovery = FALSE)                   # Reprint entire file after jam.

300 (network_address = rmtptr1.mycorp.com    # Remote text/PCL printer.
     setup_file = LJPORTRT.HPENV.SYS         # Portrait mode, etc., setup file.
     poll_interval_max = 120                 # Poll_interval decays to 2 min.
     data_timeout = 20                       # Allow for busy network.
     snmp_timeout = 20                       # Same network, same timeout.
     snmp_max_retries = 1                    # Only give it one chance.
     message_interval = 1)                   # Assures that messages are
                                             # always displayed despite
                                             # printer's distance from host.

301 (network_address = rmtptr1.mycorp.com    # Same remote printer as above.
     setup_file = LJPS.HPENV.SYS             # PostScript setup file.
     poll_interval_max = 120                 # Poll_interval decays to 2 min.
     data_timeout = 20                       # Allow for busy network.
     snmp_timeout = 20                       # Same network, same timeout.
     snmp_max_retries = 1                    # Only give it one chance.
     message_interval = 1)                   # (see comment for LDEV 300).
```

The sample printer setup files

The printer setup files described here are similar to those used in the small configuration example. To avoid repeating identical information, you are referred to the previous section where it is appropriate.

LPGLOBAL.HPENV.SYS

The global setup file LPGLOBAL.HPENV.SYS is identical to the setup file LJ4SISET used in the small configuration. Refer to that section for more information.

LJMPE.HPENV.SYS

In network printing configuration, the contents of a specific setup file are *appended* to those of any global setup file. This means that the setup data shown below, from the default MPE setup file LJMPE.HPENV.SYS, is sent *after* the setup information stored in LJGLOBAL. The only setup items that you need to include in LJMPE are those that differ from the items in LJGLOBAL.

As in the preceding example, the annotations shown here in italics are included only for clarity. Do **not** include them, or other comments, in your files.

Esc&*l*10	*Set landscape mode*
Esc(s13H	*Sets pitch to 13*
Esc)s13H	*Sets pitch for secondary character set*
Esc&a4L	*Sets left margin*

LJPORTRT.HPENV.SYS

This example assumes that the printer identified as LDEV 300 uses European standard A4 paper. As a result, the LJPORTRT setup file includes two modifications to LPGLOBAL.

Esc&*l*26A	*Selects A4 paper*
Esc&a7L	*Sets left margin*

LJPS.HPENV.SYS

This example, like the previous one, assumes that all required PostScript, including any necessary non-resident font definition, is in the user's data stream. The LJPS setup file merely switches the printer's language personality to PostScript so that it interprets the data properly.

The first line of the setup file is a Universal Exit Language command. The carriage return and line feed characters (shown as the CR and LF keycaps, respectively) in the PJL commands are significant and must be included as data. Do not rely on MPE or the spooler to supply any of these characters as line terminators. The spooler concatenates only the separate data records in the setup file and sends the result to the printer as one unbroken data stream. It does not insert any data of its own.

```
Esc%-12345X@PJL CR LF
@PJL COMMENT Beginning PostScript Job CR LF
@PJL ENTER LANGUAGE = POSTSCRIPT CR LF
```

The spooler automatically returns the printer to PCL mode at the end of each print job. There is no need for a user-supplied "exit" setup file, nor is there any provision for the spooler to use such a file.

Spooler Processes and Network Printing

The spooler uses two spooler programs, OUTSPOOL.PUB.SYS for non-network printers and OUTSPTJ.PUB.SYS for network printers. Both are similar in operation and internal interface. The main difference is the type of printers they support.

SPOOLMOM.PUB.SYS, the parent process of all spoolers, chooses the correct spooler program file for a given printer based on information available when it creates the spooler process for that printer. The choice is based on the type and subtype configuration of the printer for which a spooler process is being created. If the type/subtype is 32/0, SPOOLMOM checks the program file specified in the device's `program_file` entry in NPCONFIG.PUB.SYS. If the entry is not specified or if it is specified as OUTSPTJ.PUB.SYS, then OUTSPTJ.PUB.SYS is started as the spooler process for that LDEV. If anything else is specified, SPOOLMOM will not start a spooler process for that LDEV. If the type and subtype are anything except 32/0, SPOOLMOM runs OUTSPOOL.PUB.SYS. If the printer has been configured as initially spooled in SYSGEN, SPOOLMOM starts the correct spooler process at system start time.

You control spooler processes for both non-network and network printers with the same commands and syntax. Thus, `STARTSPOOL`, `SPOOLER <ldev>; START`, `STOPSPOOL`, `SPOOLER <ldev>; SUSPEND` all work for both types of printers. Refer to chapter 4 for detailed information about these commands.

SPOOLMOM creates OUTSPTJ processes in the CS queue (by default), just as it does for OUTSPOOL processes that manage serial printers, and for the same reason. Some large system configurations allow hundreds of printers to be connected. If these were all printing output at the same time, and were all created in the BS (linear) queue, user processes in the CS queue could be starved for CPU time. Creating spooler processes in the CS queue allows them to compete equitably with user processes for CPU time.

You may set the scheduling queue to BS, CS, DS, or ES by specifying the `run_priority` item in the NPCONFIG.PUB.SYS configuration file described earlier in the "NPCONFIG configuration file" section.

Network spooler process operation

A network spooler process operates quite similarly to a traditional spooler process. You create a spooler process by issuing a `STARTSPOOL` or `SPOOLER <ldev>; START` command, manage spooler processes by the other usual MPE/iX spooler commands, select files to print using the same mechanism as a traditional spooler process, and so on. The major difference between the two spooler processes is their device management.

A traditional spooler process manages a channel-attached printer or a serial line printer connected via a DTC. Once the printer has been allocated to that process, it is the exclusive property of the process until the process terminates. A network spooler process manages the connections to a network printer. This printer is *not* the exclusive property of the process, but must be shared among an unknown number of hosts. The network spooler process responds to this need by competing with all other hosts to connect to the printer whenever it has output to print. Whenever it fails because another host has attached the printer, it retries at configurable intervals. (See the `poll_interval` and `poll_interval_max` item descriptions in the NPCONFIG section earlier in this chapter.) The network spooler process releases the connection to the printer after each copy of each file, thus allowing other hosts a chance to access the printer.

Note that although the network spooler process releases the *printer connection* between files, it does not release or surrender the MPE LDEV. The LDEV belongs exclusively to the network spooler process, and its spool queues remain open.

Page Count Logging for network printers

The spooler creates a Spoolfile Done system log record at the end of each copy it prints. One item in this log record is a count of the total number of pages printed for that copy.

Of the currently supported devices, the HP2680, the HP-IB 256x printers, and the HP5000/F1xx series printers keep track of their actual page count and report it to the spooler at the end of the copy. Serially connected printers do not, so the spooler estimates a page count for these devices as (number of spool file records/60). Network printers with a bidirectional interface and supporting the JOB variable of PJL's USTATUS command can report an actual page count for logging. For all other printers, the spooler estimates the page count using the same method as for serial printers.

Whether the page count is actual or estimated for a particular device is shown in the "Supported Devices" section.

Operating a Network Printer

This section includes information for people using a network printer that the system manager has already configured. Such users can be a member of the system administration staff (managers, operators, etc.) or a standard MPE/iX user. The topics include:

- Using the ENV parameter to designate a setup file
- Using special forms on a network printer
- Managing perforation and page separation on network printers
- Acceptable text formats for network printers

Using the ENV parameter to designate a setup file

Any user can determine how a file is printed on a network printer by including the `ENV` parameter in a file equation or in an (HP)FOPEN intrinsic. To do so, you create the setup file and then enter its file name in the form `ENV=<filename>`.

When you specify a setup file in an `ENV` statement, the spooler assumes that it defines the entire printing environment. As a result, it supercedes all other setup file specifications in the file NPCONFIG.PUB.SYS, which the system manager created for your network environment.

If the setup file exists and if you have read access to it, the spooler opens the print file and prepends the contents of the setup file, ahead of user data, to the spool file. The information remains there for the life of the spool file.

Note

Unlike environment files for other printers, no validity checks (proper filecode, syntax, other contents, etc.) are performed for printer setup files. Be sure that you consult your printer's documentation to create a valid setup file.

Managing special forms on network printers

The spooler manages special forms requests on non-network printers as follows:

- Print a header, if headers are enabled.
- If dialog is required, prompt the operator to approve printing the file.
- If operator approves, conduct the forms alignment dialog and print the file.
- If operator does not approve, print a trailer and defer the file.

In a network printing environment, this procedure is not feasible because, once the host acquires a network printer, it should print its file without interruption and release the printer. For these reason, HP recommends that users adopt one of the two following operational strategies for handling pre-printed forms. You should either avoid the use of preprinted special forms on a network printer. Or, you should dedicate the printer to one host, so that no other host on the network can print to it.

If you do use a dedicated network printer for printing special forms, you may also want to ensure that the spooler always conducts the forms message dialog for all copies of all files that include a forms message. The spooler's default behavior is to conduct such a dialog only when the forms message of the current file is different from that of the previous file. To override the default so that the spooler conducts the dialog each time, you can use the `FORMSALIGN` command. For example, to control the dialog for LDEV 6, you would enter:

```
:FORMSALIGN 6; EACHCOPY,NOFORMIDOVERRIDE
```

Refer to chapter 4 and to the *MPE/iX Commands Reference Manual (Volumes I and II)* (32650-60115) for a complete description of the `FORMSALIGN` command.

Managing perforation and page separation on network printers

"Z-fold" paper is a continuous sheet of paper whose pages are separated by perforations which allow the pages to be folded together. Later they can be separated by tearing along the perforations. Such paper typically includes pinfeed (or tractor) holes for proper positioning in printers. By contrast, "cut-sheet" paper comes in individual sheet pages with no tractor holes.

Z-fold paper printers define an area near the perforations (typically 0.5 inch on either side) as the "perforation skip" area. Users can choose to avoid this area when printing. If they do, paper motion which causes the printer to enter this area instead causes the printer to continue advancing paper until it has spaced over the area.

MPE supports this conditional avoidance with the %102 (perf-skip on) and %103 (perf-skip off) CCTL codes. But the %2xx series of CCTL codes advance paper without regard to perforation (that is, they behave as if perf-skip is off, no matter what its actual setting). And the %3xx series of CCTL codes skip to a channel, which has no relation to the perf-skip mode. As a result, there are only a few CCTL codes, such as %60 ("0", double space) and %55 ("-", triple space), for which the perf-skip mode has any meaning.

The majority of network printers are cut-sheet printers, but many of them support the concept of top and bottom margins. As a result, there is a PCL perforation skip mode command. With perf-skip mode enabled, a Line Feed which would move the logical pen into the bottom margin area instead moves it beyond the top margin area of the next logical page. With perf-skip mode disabled, such a Line Feed allows the logical pen to occupy and print in the bottom margin area, to the physical limits of the printer.

The spooler supports enabling and disabling perforation skip mode, mapping CCTL codes %102 and %103 to their appropriate PCL counterparts, and implementing the %60 and %55 CCTL codes as two or three Line Feeds, respectively.

Acceptable text formats for network printers

This section describes the two forms in which applications can generate data destined for network printers which support only PCL, or PCL and, via PJL, the PostScript language. These forms are MPE record-oriented output or bytestream files, which may be raw PCL data, PostScript data, or ASCII text.

MPE record-oriented output

MPE record-oriented output is typically one line of ASCII text per spool file record. Each record includes a carriage control (CCTL) code that specifies how paper is to be moved vertically. Examples of vertical motion are advancing N lines, starting a new page, etc. Other MPE options allow for this motion to take place either before or after the text data (prespace or postspace mode, respectively). The default mode at the start of data is postspace mode.

Users need not include an explicit CCTL specification. If they do not, MPE provides a default that results in one `<CR><LF>` at the end of each text record.

Bytestream ASCII text files

Bytestream ASCII text files are lines of ASCII text without a record structure imposed by a file system. In bytestream files, lines are separated by the `<LF>` character and pages are separated by the `<FF>` character. The `<CR>` character may precede either of these, though it is not required. Unless the bytestream file was opened explicitly as bytestream data by a POSIX-aware application, MPE/iX would open it as an emulated variable-length file. In this case, the spooler would:

- treat a `<LF>` in the data as an end of record
- terminate an `FREAD` if the request is longer than the data preceding the `<LF>` or, if the request is shorter, discard any data between the last data byte read and the `<LF>`
- start the next `FREAD` at the byte following the `<LF>`
- pass the `<FF>` and `<CR>` characters unchanged

In contrast to bytestream files, MPE spool files have a very specific record-oriented file format. Therefore, to have the MPE spooler successfully process and print bytestream ASCII files, you must first convert them.

There are two general situations in which you are likely to need to know how to convert bytestream data for printing: when you want to print an existing bytestream file on disc, and when you are using a POSIX application that writes bytestream data *directly* to a spoolfile.

Printing a bytestream disc file. Bytestream files can exist on disc, either written directly to the HP 3000 by a POSIX application or transferred to the HP 3000 from a backup archive or another system. What method you use to successfully print a bytestream disc file depends upon whether or not the file contains motion control sequences to manipulate the *logical pen* of the printer other than simple `<LF>` and `<FF>` characters.

If the bytestream file on disc *has* motion control sequences such as PCL escape sequences or PostScript commands, use a file copy utility that (1) copies the bytestream file byte for byte, and (2) specifies a carriage control code of octal 320, which tells the spooler not to move the printer's logical pen on its own, but rather leave all such movements to the control of the file's data itself.

If the disc file has no motion control sequences other than simple `<LF>` and `<FF>` characters, use the MPE/iX `PRINT` or `FCOPY` commands (or almost any other file copy utility) to copy it to a spool file.

Writing bytestream data directly to a spool file. A POSIX application can use POSIX bytestream procedures to write directly to a device file that has the MPE/iX spool file format. The *bytestream emulator* portion of the MPE/iX file system automatically translates the bytestream data into variable length records, each with a carriage-control code. If the bytestream is simple ASCII text, such automatic translation to the spool file format will probably produce an acceptable report.

If, however, the bytestream data represents a graphic image, this automatic translation to MPE spool file format is not acceptable. In this situation, you may either:

- Modify the application so that it uses MPE/iX file system instrinsics, not POSIX procedures, to create the output spool file. In this case, the program must explicitly specify octal 320 as the carriage control character, which tells the spooler not to move the logical pen; the data stream itself will handle carriage control motion.
- Direct the application's bytestream output directly to a disc file, then use a utility to copy the disc file to spool file and specify octal 320 as the carriage control character. Again, this tells the spooler not to move the logical pen on its own.

Examples of converting bytestream files. Consider the following example, where `myfile` is a bytestream file:

```
:FILE FROM=myfile
:FILE TO;DEV=LP;CCTL
:FCOPY FROM=*FROM; TO=*TO
```

FCOPY opens file *TO with CCTL, using the configured record length of a printer in class LP as a default. Since *myfile* does not have a carriage control attribute, FCOPY specifies a CCTL code of 0 when writing a record to *TO. This code, having no special function, results in a single `<CR><LF>` by default. This effectively replaces the `<LF>` (new line character) in the original bytestream file.

Now consider a different example, where, instead of using `FCOPY` to create a spool file, a POSIX-aware application creates the file. This application could open *myfile* as a true bytestream file, then copy substrings of any size (including any `<LF>`s) to *TO specifying a CCTL of %320 on each FWRITE. When the network printer processes this file, the %320s tell it to not insert any paper motion sequences of its own. The data stream sent to the printer is then an exact replica of the original bytestream file.

4

Commands Reference

The commands that affect spooling have the following major functions:

- to control spool files
- to control spooler processes (processes operating spooled devices)

This chapter contains the complete syntax for the native mode spooler commands as well as other related commands.

The MPE/iX commands that can affect your use of the NMS are the following:

```
BUILD          OPENQ
COPY           OUTFENCE
FILE           PURGE
FORMSALIGN     RENAME
JOB            SHUTQ
LISTEQ
LISTF
LISTFILE
```

`LISTF` and `LISTFILE` are MPE/iX commands that display file information.

The native mode spooler commands are the following:

```
LISTSPF
SPOOLER
SPOOLF
```

Other MPE/iX commands that are related to NMS but are not discussed in this chapter are the following:

```
STARTSPOOL          ALTSPOOLFILE          HEADON
STOPSPOOL           DELETESPOOLFILE       HEADOFF
SUSPENDSPOOL        SHOWIN
RESUMESPOOL         SHOWOUT
```

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information about MPE/iX commands.

BUILD

Creates and immediately allocates a new empty file on disk.

Syntax

```
BUILD filereference
[ ;REC=[ recsize ] [ ,[ blockfactor ] [ , { F } ] [ , { BINARY } ] ] ]
                                        { U }       { ASCII  }
                                        { V }
[ { ;CCTL   } ] [ ;TEMP ]
{ ;NOCCTL }
[ ;DEV=[ [ dsdevice ]# ] [ device ] ]
[ ;CODE=filecode ]
[ ;DISC=[ numrec ] [ ,[ numextents ] [ ,[ initialloc ] ] ] ]
[ { ;RIO   } ]
{ ;NORIO }
[ [ { ;MSG    } ] ] [ ;ULABEL=numlabels ]
  { ;CIR    }
  { ;STD    }
  { ;KSAMXL }
  { ;SPOOL  }
[ ;KEY={ ^filereference } ] [ ;FIRSTREC=recnum ] [ { ;REUSE   } ]
       { keyinfo       }                           { ;NOREUSE }
[ ;langid={ langid   } ]
          { langname }
```

Enhancements

filecode

The *filecode* parameter specifies a particular kind of file. This code is recorded in the file label and is available to processes accessing the file through the FFILEINFO or FGETINFO intrinsics. Although you can specify a positive integer ranging from 0 to 32767 or a mnemonic name for *filecode*, certain reserved integers and mnemonics have particular system-defined meanings. The native mode spooler file codes are the following:

Integer	Mnemonic	Meaning
1515	INSP	input spool file
1516	OUTSP	output spool file
1517	CHKSP	output spool file checkpoint file

`SPOOL` `SPOOL` specifies an output spool file that is not linked to the spool file directory (`SPFDIR`) and, therefore, is not printed automatically. No spooling attributes are initialized.

If the output spool file is ever linked to the `SPFDIR` by using the `SPOOLF ... ;PRINT` command, all attributes are set at that time according to the rules of the command. Spool Files cannot be temporary files. If you specify the `;SPOOL` keyword, a file code of 1516 (output spool file) is forced.

COPY

Copies one file to another by creating a new file or by overwriting an existing file. **(Native Mode)**

Syntax

```
COPY [FROM=]sourcefile[{;TO=}]targetfile[{ASK}]
                     [{,   }]           [{YES}]
                                        [{NO }]
```

Enhancements

The `COPY` command allows the copying of nonprivate output spool files. The new file is not linked to the spool file directory (`SPFDIR`).

FILE

Declares the file attributes to be used when a file is opened. This declaration, informally known as a file equation, may be used to override programmatic or system default file specifications. With the addition of shared parameters from the NS3000/XL AdvanceNet subsystem, the declaration may specify a formal file designator that may be used to access a remote file or device in a subsequent command or intrinsic. NS3000/XL AdvanceNet is not part of the 900 Series HP 3000 fundamental operating system and must be purchased separately.

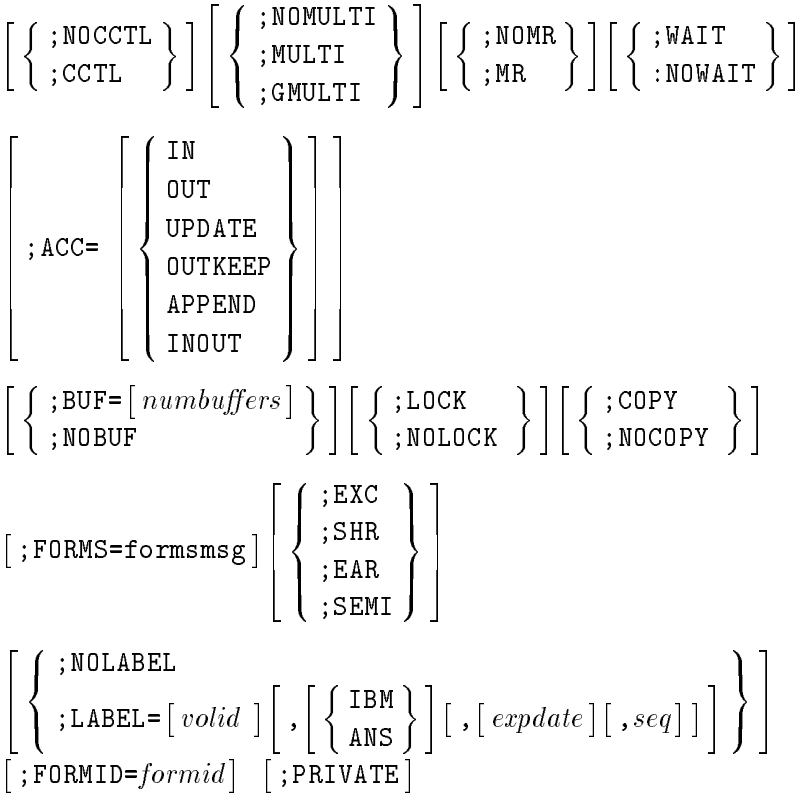
Syntax

```
FILE formaldesignator=*formaldesignator
                      =$NULL
                      =$NEWPASS
                      =$OLDPASS
                      =$STDIN
                      =$STDINX
                      =$STDLIST

=filereference[ :nodespec ]
              [ ,filedomain ]

[;DEV=[[envname]#][device][,outpri][,numcopies]]

[;VTERM][;ENV=envfile[:nodespec]]

[;option][;access][;disposition]
```

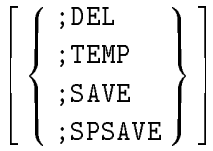
Syntax for option

```
;REC=[recsize][,[blockfactor][,{F}][,{BINARY}]]
                               {U}  {ASCII }
                               {V}

;DEN=[density];DISC=[numrec][,numextents][,initialloc]

;CODE=[filecode][{;RIO  }][{;STD    }]
                {;NORIO}  {;MSG    }
                          {;CIR    }
                          {;KSAMXL }
                          {;SPOOL  }

[;ULABEL=numlabels]  [;KEY={^filereference2}]
                           {keyinfo        }

[;FIRSTREC=recnum]  [{;REUSE  }]
                     {;NOREUSE}

[;LANG={langid  }]
       {langname}
```

Syntax for access



Syntax for disposition



Enhancements

filecode

The *filecode* parameter specifies a particular kind of file. This code is recorded in the file label and is available to processes accessing the file through the FFILEINFO or FGETINFO intrinsics. Although you can specify a positive integer ranging from 0 to 32767 or a mnemonic name for *filecode*, certain reserved integers and mnemonics have particular system defined meanings. The native mode spooler file codes are the following:

Integer	Mnemonic	Meaning
1515	INSP	input spool file
1516	OUTSP	output spool file
1517	CHKSP	output spool file checkpoint file

SPOOL

The SPOOL option specifies an output spool file that is not linked to the spool file directory (SPFDIR) and, therefore, will not be printed automatically.

No spooling attributes are initialized. If the output spool file is ever linked to the SPFDIR by using the SPOOLF ... ;PRINT command, all attributes are set at that time according to the rules of the command. If you specify ;SPOOL, a file code of 1516 (output spool file) is forced.

formid — The *formid* parameter applies only to output spool files. It is a string of up to 8 alphanumeric characters, beginning with a letter, that uniquely identifies a special form to mount. Refer to chapter 2 for a discussion about *formid*.

PRIVATE — The PRIVATE option generates a spool file that can be accessed in privileged mode only. Private spool files may not be saved or copied. They may only be purged, printed, or (within limits) altered by using the SPOOLF command. The PURGE and COPY commands do not work on private spool files. Refer to the SPOOLF command in this chapter and to chapter 1 for information about private files.

SPSAVE — If you use this parameter, the output spool file is not purged after the last copy of it has printed. The OUT.HPSPOOL account retains the spool file. You cannot use the SPSAVE parameter with a private spool file.

FORMSALIGN

Configures one spooled printer, or a group of spooled printers related by device class, to conditionally enter into a forms message dialog with its operator(s) when the current spool file includes a forms message.

Syntax

```
FORMSALIGN [DEV=] { ldev | devclass | devname }

[ ;[DIALOG=] { { EACHCHANGE | EACHFILE | EACHCOPY } [ , { FORMIDOVERRIDE | NOFORMIDOVERRIDE } ] } ]

[ ;SHOW ]
```

Parameters

ldev — This is the logical device number of a printer. The printer must be configured as an MPE type 32 device.

devclass — This is the device class name of a class of printers. Each printer in the class must be configured as an MPE Type 32 device. The *devclass* name must begin with a letter and consist of eight or fewer alphanumeric characters.

devname — This is the device name of a printer. The *devname* must begin with a letter and consist of eight or fewer alphanumeric characters. It is not possible to have a *devclass* name and a *devname* name that are the same. If you enter an alphanumeric character string, the command searchs the device class list first, and then the device name list.

`EACHCHANGE` — The spooler process conducts the forms message dialog only when these two conditions are met:

- when the (case-insensitive) forms message of the current spool file differs from that of the previous spool file printed by that process *and*
- when an overriding formid specification is not in effect

Two different spool files (different SPOOLIDs) with the same forms message will print without the forms message dialog if they are printed consecutively.

`EACHFILE` — The spooler process conducts the forms message dialog only when these two conditions are met:

- when the spoolid of the current spool file differs from that of the previous spool file printed by that

process, the current spool file contains a forms message *and*

- when an overriding formid specification is not in effect

The second and subsequent copies of the same spool file will print without the forms message dialog if they are printed consecutively.

`EACHCOPY` The spooler process conducts the forms message dialog for every copy of every spool file that contains a forms message *if* an overriding formid specification is not in effect.

`FORMID-OVERRIDE` This is a subparameter of the chosen `EACH`*xxxx* keyvalue. With this feature selected, the native mode spooler first checks its current and previous spool files for the same nonblank, case-insensitive FORMID. If the FORMIDs match, both the `DIALOG` option for the spooler process and any forms message in the current spool file are ignored, and no forms message dialog takes place. In other words, identical FORMIDs override all other considerations.

Note that the `DIALOG` option is not changed. It is ignored as long as the two FORMIDs match.

If the two FORMIDs do not match, the spooler conducts the forms message dialog using the forms message of the current spool file. If the current spool file has no forms message (whether or not it has a FORMID), the spooler:

- conducts no dialog if standard forms are already mounted
- conducts the standard forms dialog if special forms are mounted

Any `DIALOG` option is ignored.

`NOFORMID-OVERRIDE` This is a sub-parameter of the chosen `EACHxxxx` keyvalue. With this feature selected, the native mode spooler ignores any and all FORMIDs associated with the current spool file or the previous spool file. The setting of the `DIALOG` option always determines the conditions under which the spooler process conducts the forms message dialog. The FORMID is then useful only as an item in a selection equation.

The setting of (`NO`)`FORMIDOVERRIDE` only affects the spooler's behavior with respect to the forms message dialog. It has no effect on the use of the `FORMID` keyword in a selection equation of either the `SPOOLF` or `LISTSPF` command. Thus, it is still possible to

select a subset of all spool files to alter, delete or display, based on FORMID=, regardless of the setting of (NO)FORMIDOVERRIDE for a given device. They are independent of each other.

Note

No matter which set of the above options is selected, if the current spool file has no forms message but special forms are mounted on the device, the spooler conducts the standard forms dialog.

If the DIALOG option is omitted, the configuration is not changed.

SHOW — Specifying this option causes the configuration for the specified DEV(s) to be displayed as in the examples below. When ;SHOW is used alone, the current configuration is displayed. If other parameters are present, the configuration is first updated and then displayed.

If this option is omitted, nothing is displayed.

Operation notes

The FORMSALIGN command can be used on a spooled or an unspooled printer, or on a device class containing any mixture of spooled and unspooled printers. When used on a spooled printer, the specified options become effective on the next copy selected for printing on that device. The choices are retained until changed by another FORMSALIGN command, even if the printer should become unspooled and a new spooler process started for it.

When used on an unspooled printer, it has no effect but will be retained (unless changed by another FORMSALIGN command) and will become effective immediately upon spooling the printer. Files that include a forms message and that are directed to an unspooled printer always trigger a forms message dialog with the printer's operator. Any FORMID accompanying the file is irrelevant when the file is directed to an unspooled printer.

The options specified in the FORMSALIGN command are stored in the appropriate device files. For example, options for LDEV 6 are stored in file 00000006.DEVICES.3000devs. This is why the options are retained even when no spooler process exists for LDEV 6.

There is one file name entry in this group for every configured device on the system. These files contain information vital to the internal management of all configured devices. A corresponding CLASSES group in the 3000devs account holds one file name entry in this group for every configured class on the system.

These device files are reconstructed at each system startup, and options other than the default (EACHCHANGE, FORMOVERRIDE) are not retained across system startups. Thus, your SYSSTART file should include one FORMSALIGN command per device or class for which you want to set options other than the default.

Note

Because this command may affect more than one device (if applied to all devices in a class), it is possible to get warnings for some of those devices and not for others. A warning on one or more devices affects only that device. The command continues to execute until all selected devices have been configured and/or shown, or an error is detected. An error terminates the command.

A sample of the output might be:

```
FORMSALIGN LP;SHOW

                                    FORMID
 LDEV   DEVNAME    DIALOG          OVERRIDE

    6   LDEV6      EACHCHANGE      YES
   14   LDEV14     EACHCOPY        NO
   15   LDEV15     EACHFILE        YES
   19   LDEV19     EACHCHANGE      NO
```

Example 1

Current (pre-command) native mode spooler operation:

1. Check the FORMIDs of the previous and current spool file.
2. If they are both nonblank and identical, skip any forms message dialog.
3. If they are different, conduct a dialog. Determine the proper dialog by examining the forms message of the current spool file.
   i. If the forms message is nonblank (that is, it was explicitly specified by a user when the spool file was created), enter into the normal forms message dialog.
   ii. If the forms message is blank and special forms are mounted, prompt the operator to mount standard forms.
4. If the FORMIDs of the previous and current spool file are both blank:
   i. Examine the forms message of the current and previous spool files.
   ii. If they are the same, skip any forms message dialog.
   iii. If they are different:
      1) If the current forms message is nonblank, enter into the normal dialog.
      2) If it is blank, prompt for standard forms.

To duplicate this operation, enter:

```
FORMSALIGN 6; DIALOG=EACHCHANGE, FORMIDOVERRIDE
```

Example 2 The MPE V/E and CM spoolers conducted a forms message dialog for each copy of each spool file printed. To duplicate this operation, enter:

```
FORMSALIGN 6; DIALOG=EACHCOPY, NOFORMIDOVERRIDE
```

Examples 3 and 4 assume that LDEV 6's spooler process prints the sequence of files shown below, and that the sequence starts with standard forms on the device. The FORMID and FMSG columns denote actual text except for "(none)", that denotes the absence of text. Example 3 assumes that FORMIDOVERRIDE is in effect, while example 4 assumes that NOFORMIDOVERRIDE is in effect. For each example, a mark under a DIALOG option indicates that the spooler conducts a normal forms message dialog (F) or prompt for standard forms (S) at that point in the sequence with that DIALOG option in use.

Example 3 FORMIDOVERRIDE

SEQ #	SPOOLID	FORMID	FMSG	COPY #	EACHCHANGE	EACHFILE	EACHCOPY
1	0100	(none)	MSG01	1	F	F	F
2	0100	(none)	MSG01	2			F
3	0101	(none)	MSG01	1		F	F
4	0102	(none)	(none)	1	S	S	S
5	0102	(none)	(none)	2			
6	0103	FORM02	MSG02	1	F	F	F
7	0103	FORM02	MSG02	2			
8	0104	FORM02	MSG03	1			
9	0105	FORM02	(none)	1	S	S	S
10	0106	FORM00	(none)	1			
11	0107	FORM04	MSG04	1	F	F	F
12	0108	FORM05	MSG04	1	F	F	F
13	0108	FORM05	MSG04	2			
14	0109	(none)	MSG04	1	F	F	F
15	0110	(none)	(none)	1	S	S	S

Notes (by SEQ #):

1 No FORMID, and the forms message text changes. Here the assumption is made that we started with standard forms.

6 Since FORMIDOVERRIDE is in effect, the change in FORMID from SEQ 5 to 6 triggers the forms message dialog regardless of which DIALOG option is in effect.

7 Since FORMIDOVERRIDE is in effect, the same FORMID as in SEQ 6 overrides an EACHCOPY specification.

8 In general, avoid using the same FORMID on files with different forms message texts. As in this example, the options causes the forms message,

MSG03, to be skipped. That may not be the effect you intend.

10 The standard forms dialog is not repeated, even with a new FORMID.

12 Here the FORMID changes from SEQ 11 but the forms message does not. Since we are using FORMIDOVERRIDE, the FORMID change triggers the forms message dialog.

14 Same as note 12, except that the FORMID change is from an explicit FORMID to no FORMID. Because the forms message is nonblank, the spooler enters into a normal forms message dialog. Had the forms message been blank, the spooler would have prompted the operator for standard forms here instead of at SEQ 15.

Example 4 NOFORMIDOVERRIDE

SEQ #	SPOOLID	FORMID	FMSG	COPY #	EACHCHANGE	EACHFILE	EACHCOPY
1	0100	(none)	MSG01	1	F	F	F
2	0100	(none)	MSG01	2			F
3	0101	(none)	MSG01	1		F	F
4	0102	(none)	(none)	1	S	S	S
5	0102	(none)	(none)	2			
6	0103	FORM02	MSG02	1	F	F	F
7	0103	FORM02	MSG02	2			F
8	0104	FORM02	MSG03	1	F	F	F
9	0105	FORM02	(none)	1	S	S	S
10	0106	FORM00	(none)	1			
11	0107	FORM04	MSG04	1	F	F	F
12	0108	FORM05	MSG04	1		F	F
13	0108	FORM05	MSG04	2			F
14	0109	(none)	MSG04	1		F	F
15	0110	(none)	(none)	1	S	S	S

Notes (by SEQ #):

all: Dialog decisions are based exclusively on forms message changes and the DIALOG option, because NOFORMIDOVERRIDE is in effect.

8 The caveat of example 3 is not as important here, since any FORMID or lack of FORMID does not contribute to the forms message dialog decision.

Use

This command may be issued from a session or a job, in a break, or in a program. It is not breakable while updating the configuration, but is breakable during the display portion of the command if the ;SHOW option is used. Any user may execute the command with the ;SHOW option alone, in order to display current configurations. When changing a configuration, it may be executed from the console or by a user to whom the command has been allowed with the ALLOW command or by a user who has used the ASSOCIATE command on a *devclass* or a device class that includes *ldev* or *devname*.

Errors

The FORMSALIGN command may generate this warning message:

```
DEVICE "\" MUST BE A PRINTER. (CIWARN 4627)
```

The FORMSALIGN command displays this message if *ldev* or *devname* is not configured as an MPE Type 32 device (a printer). If the *devclass* form of the command is used, this message is displayed for each LDEV in *devclass* that is not a printer.

A logical device number is substituted for the "\" shown above.

JOB

Defines a job to be scheduled with the STREAM command or an input spooled device to run in batch mode.

Syntax

```
JOB [jobname,]username[/userpass].acctname[/acctpass]
[,groupname[/grouppass]]

[;TIME=cpusecs][;PRI={BS|CS|DS|ES}]

[{;INPRI=inputpriority|;HIPRI}][;RESTART]

[;OUTCLASS=[device][,outputpriority[,numcopies]]]

[;TERM={termtype}]

[;PRIVATE]

[;SPSAVE]
```

Enhancements

PRIVATE The PRIVATE option generates a spool file that can be accessed in privileged mode only. Private spool files may not be saved or copied. They may only be purged, printed, or (within limits) altered by using the SPOOLF command. The PURGE or COPY commands do not work for private spool files. Refer to the SPOOLF command in this chapter and to chapter 1 for information about private files.

SPSAVE If you use this parameter, the output spool file is not purged after the last copy of it has printed. The OUT.HPSPOOL account retains the spool file. You cannot use the SPSAVE parameter with a private spool file.

LISTEQ

Displays all active file equations for a job or session.

Syntax

`LISTEQ` [*listfile*]

Enhancements

`LISTEQ` displays the parameters `PRIVATE` and `SPSAVE`.

LISTF

Displays information about one or more permanent files. `LISTF` does not display file information for files residing in hierarchical directories. To list such files, use the `LISTFILE` command. Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information on listing hierarchical files.

Syntax

`LISTF` [*fileset*] [,*listlevel*] [;*listfile*]

Parameters

fileset — Specifies the set of files to be listed. The default is `@`, producing a listing of all files in the logon group. You may select the file(s) to be listed by using the fully or partly qualified form for *fileset*:

filename`[.`*groupname*`[.`*accountname*`]]`

Use the `#` symbol to specify a single numeric character. Use the `?` symbol to specify a single alphanumeric character. Use the `@` symbol to specify zero or more alphanumeric characters. By itself, `@` represents all the members of a set. Each of these wildcard characters counts toward the eight-character limit for group, account, and file names.

listlevel — Specifies the level (amount and format) of information about the file(s) you select. The default is zero.

-3 — Displays the same information found with 3, plus the lockword, creator, and label address. For program files the `#SEG`, `STACK`, `MAXDATA`, `TOTAL`, `DB`, `DL`, and `CAP` values are omitted. System manager (SM) or account manager (AM) capability is required to use this option.

-2 — Displays the file's access control definition (ACD). System managers can view the ACD for any file. Account managers can view the ACD for files in that account. File creators can view the ACD for their files. Other users can view an ACD only if that ACD specifies that the user has RACD (read ACD) access.

-1 — Displays the hexadecimal listing of the file label, including all lockwords. This level is available only to system managers and account managers.

Note

The hexadecimal listing generated by a `LISTF, -1` serves only a diagnostic purpose in MPE/iX and is subject to change.

0 — Displays only the file name. This is the default.

1 — Displays the file name, file code, record size, record format, and other file characteristics such as ASCII

	or binary records, carriage-control option, current end-of-file location, and the maximum number of records allowed in the file.
2	Displays the file name, file code, record size, record format, and other file characteristics such as ASCII or binary records, carriage-control option, current end-of-file location, and the maximum number of records allowed in the file. It also displays the blocking factor, number of sectors in use, number of extents currently allocated, and the maximum number of extents allowed. LISTF, 2 also displays KSAMXL file types with "K".
3	Displays the file name, record size, extent size, number of records, access rights for the user, and other file characteristics including the date created, modified, and last accessed. The creator, lockword, and label address are omitted. These can be obtained by specifying -3 if you have AM capability (for files in your account) or SM capability (for any file on the system).
4	Displays the security matrix for the file. This includes account, group and file-level security and the access rights for the user. If an access control definition (ACD) exists, a message stating that fact is displayed.
6	Displays the fully qualified file name.

LISTFILE

Lists file information. The `LISTFILE` command displays file information for files residing in hierarchical directories. For more information about the `LISTFILE` command, refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)

Note

Spool files, which reside in `IN.HPSPOOL` or in `OUT.HPSPOOL`, are named according to MPE conventions and appear in a hierarchical listing only to the extent that all MPE files do so.

.

Syntax

```
LISTFILE [fileset
          (fileset [,fileset ][, ... ]) ]
[ [;FORMAT=]format_opt]  [ [;SELEQ=]select_eq]
             [   { PERM     } ]   [;USENAME]
[;PASS]      [ ; { TEMP     } ]   [;TREE   ]
             [   { PERMTEMP } ]   [;NOTREE ]
```

Parameters

fileset

Specifies the set of files to be listed.

If *fileset* does not begin with a dot or a slash, it is parsed according to the MPE syntax and has the form:

filename`[.`*groupname*`[.`*accountname*`] ]`

If *fileset* begins with a dot (`.`) or a slash (`/`), it is parsed according to the Hierarchical File Syntax (HFS). Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115).

Wildcards may be used. Patterns are the same as those for `LISTF` and `SHOWVAR`. "`[a-dq]#x`" means search for all of files beginnning with a, b, c, d, or q followed by any number followed by x. Default is @.

format_opt

A format selection. This parameter has no effect on the files selected for display. The following numbers/mnemonics are recognized:

Format Selection

Option	Name	Displayed Information
−3	(DETAIL)	Shows LISTF, 3 data plus the lockword, creator, and label address. AM or SM capability required.
−2	ACD	Shows only the access control definition (ACD) .
−1	LABEL	Shows only the file label in hex.
0	FILES	Shows only the file name.
1	SUMMARY	Shows LISTF,1 data.
2	DISC	Shows LISTF,2 data.
3	DETAIL	Shows LISTF,3 data.
4	SECURITY	Shows LISTF,4 data.
5	DATA	Shows LISTF,3 data and all file specific data in LISTF, 3 type format (KSAM and SPOOL).
6	QUALIFY	Shows only fully qualified file name.
7	UNIQUE	Shows all file-specific data in LISTF,3 type format, but does not show LISTF,3 data.

Format options 5 and 7 are "data driven" outputs that show file specific information such as KSAM keys, or target print devices.

When you use option 5 and a file has no unique data, only the `LISTF,3` data is shown. When you use option 7 and a file has no unique data, only the file name is displayed. Default = 0.

select_eq A selection equation. Use the selection equation to filter the *fileset*. From the set of files matching the file set, only files that match the filter requirements are listed. Valid selection equations may only select on file types by using the `FTYPE` parameter and may only use the "=" operator. Selection equations have the following format:

select_eq ::= [`FTYPE = mnemonic`]

`mnemonic::=` { `KSAMXL` | `SPOOL` }

Selection equations must be surrounded by square brackets.

`PASS` An option that is used to refer to sensitive data. The use of this option depends on your access rights to the data.

PERM	An option that is used to display permanent files only. This is the default.
TEMP	An option that is used to display temporary files only.
PERMTEMP	An option that is used to display both permanent and temporary files. Temporary files are listed after the permanent files.
USENAME	Applies only to Hierarchical File Syntax (HFS) named files. This option indicates that the name is to be used to determine how many levels to display. If the *fileset* ends in a slash, then all the lower level objects (based on *seleq*) are to be displayed. If the name does not end in a slash, the only the objects at the specified level are displayed. For example, /@/@/@ indicates that all objects at the third level are to be displayed. This is the default Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information on listing hierarchical files.
TREE	If TREE is specified, objects at all lower directory levels are displayed. This is the only way to have all levels displayed if the fileset is in MPE syntax.
NOTREE	If NOTREE is specified, only objects at the specified level are to be displayed. NOTREE overrides an HFS *fileset* that ends in a slash. Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information on listing hierarchical files.

Operation notes

The LISTFILE command is a functional superset of the LISTF and LISTTEMP commands. Unlike the LISTF command, the LISTFILE command supports standard native mode scanning/parsing that can be easily expanded. Instead of using difficult to remember numbers, mnemonic keywords and options are supported.

This command lists descriptions of one or more disk files at the level of detail that you select. You must have Traverse Directory (TD) entries and/or Read Directory (RD) entries for the directories in the pathname of the files that will be displayed by LISTFILE. If you are a standard user, you may list any level of information on files that you create, but you may not use the ;PASS options on files that you do not own. If you have AM capability, you may use the ;PASS options for any file within the account. If you have SM capability, you may use the ;PASS options for any file on the system. A file description is not listed unless the file's home volume set (PV) is mounted.

Use

This command may be issued from a session or a job, in program, or in BREAK. It is breakable. (You may abort its execution.)

If the *fileset* is in MPE syntax, `LISTFILE` will not display any directories, or any files that do not follow MPE naming syntax (`LISTFILE @,2`, for example) will not display the file "am_pm" created by some HFS application; however, `LISTFILE ./@,2` will. Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for more information on listing hierarchical files.

If the *fileset* is in HFS syntax and it ends in a slash (or you specify the `TREE` option), all the nodes (files and directories) are found that match the *fileset* parameter (horizontal cut). The domain of further traversal is limited to the sub-trees root at these nodes. Those files that match the pattern of *pattern* are the ones that are finally displayed using the *format_opt* specified.

If the HFS syntax *fileset* does not end in slash (or you specify the `NOTREE` option), all the nodes (files and directories) that match the pattern of *fileset* (horizontal cut) and the pattern of *pattern* are displayed using the *format_opt* specified.

In both cases, a final filter of `SELEQ` is applied, if it is present, to further restrict the files/directories to be displayed.

Example 1

```
LISTFILE SPRLRFMT, 5
********************
FILE: SPLRFMT.SPLR.DEV


FILE CODE: 1516      FOPTIONS: ASCII,VARIABLE,NOCCTL,SPOOL
BLK FACTOR: 1        CREATOR : **
REC SIZE: 1008 (BYTES) LOCKWORD: **
BLK SIZE: 1024 (BYTES) SECURITY--READ     : ANY
EXT SIZE: 39 (SECT)              WRITE    : ANY
NUM REC: 38                      APPEND   : ANY
NUM SEC: 16                      LOCK     : ANY
NUM EXT: 1                       EXECUTE  : ANY
MAX REC: 38                    **SECURITY IS ON
MAX EXT: 1            FLAGS   : NO ACCESSORS
NUM LABELS: 0         CREATED : THU, JAN 26, 2989, 3:35 PM
MAX LABELS: 0         MODIFIED: THU, JAN 26, 2989, 3:35 PM
DISC DEV #: 17        ACCESSED: TUE, MAR 14, 1989, 9:09 AM
CLASS     : DISC      LABEL ADDR: **
SEC OFFSET: 0

TARGET DEVICE : 6
```

Example 2

```
LISTFILE SPLRFMT, 7
********************
FILE: SPLRFMT.SPLR.DEV


TARGET DEVICE : 6
```

More examples

The following are more LISTFILE examples:

```
listfile x@,data,[ftype=spool]

listfile [a-f]#[g-z@],3;seleq=[ftype=spool]
```

Note

For examples of LISTFILE displays with HFS files. Refer to *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115).

LISTSPF

Produces a listing of spooled files, both input and output. (**Native Mode**)

Syntax

```
LISTSPF [ [IDNAME=] { spoolid                          } ]
        [           { (spoolid [ ,spoolid] ... )       } ]

[ [;SELEQ=] { select-eq      } ] [ ;DETAIL ]
[           { ^indirect_file } ] [ ;STATUS ]
```

Parameter definitions

spoolid

One or more spool file IDs: #I*nnn* for input spool files or #O*nnn* for output spool files. These IDs are assigned by the spooling subsystem at spool file creation time. The # is optional; but if it is used, an O or I must also be used. If it is not used, the O is also optional for output spool files; that is, if neither [#]O nor [#]I is specified, then [#]O is assumed.

The symbol `@` may be used to specify all spool files.

The symbol `O@` may be used to specify all output spool files.

The symbol `I@` may be used to specify all input spool files.

If `@`, `O@`, or `I@` is specified, it must be the only *spoolid* value supplied. `@, O@`, and `I@` are mutually exclusive.

A user with SM or OP capability or a console user who specifies O@ sees all output spool files on the system. A user with AM capability who specifies O@ sees all output spool files created by users in the same account. All other users are limited to files they have created. Similar rules apply to I@ and @. The default is all the output spool files created by the current *user.account*. The default *spoolid* for the console user is all the output spool files on the system.

Note

If you specify duplicate *spoolid*s, the system displays a warning message.

select-eq

The selection equation is used as a filter on the set of spool files selected. Only spool files whose attributes satisfy all filter requirements is listed. For example, you use the following command to display all the output spool files from *user.acct* that have less than 100 pages:

```
LISTSPF 0@;SELEQ=[(OWNER=user.acct)AND(PAGES<100)]
```

If you are not an SM, OP, AM, or console user, the following command displays all the output files in your default group with a priority greater than 2 that were created before September 30, 1989.

```
LISTSPF
0@;SELEQ=[(PRI>2)AND(DATE<09/30/89)]
```

Selection equations have this format: (In this display, interpet ::= as "can be replaced by".)

select-eq : := [*equation*]

Begin and end a selection equation with square brackets ([and]).

equation : := { *parm* { > | >= | < | <= | <> | = } *value* | (*equation*) | NOT *equation* | *equation* { AND | OR } *equation* }

The logical operator AND takes precedence over the logical operator OR. For example:

```
LISTSPF 0@;SELEQ=[FILEDES=REPT OR OWNER=BOB.ACCTG AND
PRI>8]
```

`[FILEDES=REPT OR OWNER=BOB.ACCTG AND PRI>8]` is the same as `[FILEDES=REPT OR (OWNER=BOB.ACCTG AND PRI>8)]`.

value ::= Appropriate values per data type. For example, `STATE=READY` or `PRI>6`.

parm ::= The parameter (*parm*) may be one of several attributes of the spool file to be used as filters. The *parm* choices are described in a list below.

Note

For string types other than `DATE`, such as user name, only the relational operators "=" and "<>" apply. Using any others results in an error.

- *parm ::=*`DEV`: LDEV number, device name, or device class name. You can use wildcards for device name and device class name.

- *parm ::=* FILEDES: Formal or actual file designator for the spool file. For example, if you enter the file equation below and print to it, EPOCLONG becomes the spool file's FILEDES.

```
FILE EPOCLONG;DEV=EPOC;ENV=LPLONG.ENV.SYS
PRINT MYFILE,*EPOCLONG
```

  You may use wildcards.

  This keyword supports selection on the null string by entering FILEDES= "" (You may also use single quotes). You must include such a construct if you specifically want to select such an attribute. Note that "" is not the same as " "; the blank is significant.

- *parm ::=* SPOOLID: Spool File identifier number in the format #O*nnn* or #I*nnn*.

  The "#" is optional; but if it is used, an O (for output) or an I (for input) must also be used. If # is not used, the O is also optional for output spool files; that is 123 is the same as #O123. The valid range of *spoolid*s is $1 \leq nnn \leq 9{,}999{,}999$. (The commas are for clarity; do not enter any commas in the actual equation.)

- *parm ::=* PAGES: Number of pages in the spool file (if known). A positive integer number is expected.

Note

This attribute does not apply to input spool files; therefore, any logical *condition* involving the attribute always returns FALSE when tested against an input spool file.

- *parm:=* FORMID: Form name. You can use wildcards. (The *formid* is an ASCII string up to 8 characters, the first of which must be a letter.). Refer to the notes accompanying the FILEDES and PAGES description.

- *parm:=* STATE: READY, ACTIVE, OPEN, CREATE, PRINT, PROBLM, DELPND, SPSAVE, DEFER, XFER.

- *parm ::=* JOBNAME: Job or session name under which the spool file was created. The job name can consist of up to 8 alphanumeric characters, the first of which must be a letter.

  For a job input spool file, the JOBNAME shown is allocated to that job, *not* the job or session that streamed it.

  You may use wildcards.

- *parm ::=* DISP: Disposition: SPSAVE or PURGE. Refer to the notE accompanying the PAGES description.
- *parm ::=* COPIES: Number of copies. Minimum is 1, maximum is 65,535. Refer to the note accompanying the PAGES description.
- *parm ::=* PRI: Output priority minimum is 0, maximum is 14.refer to the note accompanying the PAGES description.
- *parm ::=* JOBNUM: Job or session number under which the spool file was created, for example: #S257, #J329, or S*n* (the "#" is optional) where $1 \leq n \leq 16{,}383$. (The commas are shown for clarity; do not enter any commas in the actual equation.)

  For a job input spool file, the JOBNUM shown is allocated to the job, *not* the job or session that streamed it.

  You may use some wildcards; J@ accepts all jobs, S@ accepts all sessions. J'@ and S'@ are also allowed, The apostrophe (') indicates an imported spool file or a spool file recovered during START NORECOVERY.
- *parm ::=* RECS: Number of records in the spool file. A positive integer is expected.
- *parm ::=* OWNER: The user under which the spool file was created. The format of the *owner* is *user.account*. If the account is not specified, the user's current account is assumed. You can use wildcards.

  For a job input spool file, the OWNER is the user logon for the job, *not* the job or session that streamed it.
- *parm ::=* JOBABORT: Select based on whether or not this is the $STDLIST of a job that aborted when an error was encountered but no CONTINUE was in effect.

  Valid values are TRUE and FALSE. Only "=" and "<>" are allowed as relational operators. Refer to the note accompanying the PAGES description.
- *parm ::=* DATE: Creation date in the format *mm/dd/yy* or *mm/dd/year*. Note that the year can be in the form of *yy*, as in 10/10/88, or in the form of *year*, as in 10/10/1988; both are legal syntax for the DATE parameter.

^*indirect_file* The *indirect_file* parameter specifies the name of a file containing the selection equation. It must be preceded by a caret (^). The selection equation contained in the file may not exceed 277 characters in length, including the brackets in which it must reside. There is no restriction on the indirect file code. If the record size exceeds 277, only 277 characters per record are read and a warning is issued. Backreferencing to a formal file designator is also allowed for an *indirect_file* name; that is, ^**filename* is allowed. Any file is accepted as an *indirect_file*, unless the file system returns an error from FOPEN or FREAD.

Note

There is no limit to the number of records in the *indirect_file*, only the total character count.

Records are processed as follows:

- Leading and trailing blanks are stripped.
- If the last nonblank character is an ampersand (&), it is also stripped; otherwise, one blank is added back to the end of the record as a delimiter.
- The character count of the record is added to that of the records processed previously. If the total character count exceeds 277, an error is returned. If the total is less than 277, the current record is appended to previous records.
- This process repeats until either 277 characters have been counted or the end-of-file is detected. Records terminating with or without ampersands may be mixed as desired in the indirect file.
- If the resulting string is $\leq$277 characters, it is parsed.
- If the parser detects a syntax error, or if any nonblank character follows the closing bracket (]) of the *select-eq*, an error is returned and the *select-eq* is not processed.

DETAIL Produces a two-line description of the specified spool file(s). The default is a one-line display (not detailed).

STATUS By default, LISTSPF displays a listing of selected spool files, followed by a statistical summary of those spool files, known as the **status display**.

Specification of the STATUS option causes only the status summary to be displayed summarizing the specified file set. STATUS and DETAIL cannot be specified together.

Operation notes

This command is provided to enable users to obtain a list of spool file information without having to look for it within a list that includes other files.

The display for LISTSPF is different from the SHOWIN/SHOWOUT display. LISTSPF displays both output and input spool files. The display shows output spool files, then input spool files, and finally a summary status display.

The parameters are divided into three groups: selection, detail and status.

The selection group allows a user to limit the display of spool files to a subset of the overall group of spool files on the system.

The detail parameter displays more than the default information on the files that have been selected.

The status parameter displays summary status only.

These parameters can be combined as desired.

This command displays status information for one or more spool files. The information reflects the status at the time the command is entered and always appears on the standard list device. Within device or device class, READY, CREATE, PRINT, and XFER state output spool files are displayed first, sorted by priority and then by date and time. Output spool files in DEFER, PROBLM, or SPSAVE states are shown next sorted by order of state and then priority and time.

Output spool files are displayed first, followed by input spool files and the status display. The display for input spool files is not sorted.

Use LISTSPF when all you need is a listing of spool files.

SPOOLF 0@;SHOW, for example, may lock the SPFDIR and JMAT tables for extended periods, during the execution of the command.

When many spool files exist, this can result in lengthy delays in vital user-initiated actions.

LISTSPF does not incur lengthy delays.

Refer to the "Operation" discussion of the SPOOLF command.

Display field and description:

An example of the first line of the display for LISTSPF is:

```
SPOOLID  JOBNUM  FILEDES  PRI  COPIES  DEV   STATE   RSPFN  OWNER

#O1      J12345  $STDLIST   6       1  EPOC  CREATE  RSPFN  THISUSER.ACCOUNT1
```

The fields of the display for LISTSPF are described below.

SPOOLID	The unique spool file identifier.
JOBNUM	The job or session identifier of the job or session that created the spool file. The exception to this is that the *jobnum* for a JOB input spool file is the job number assigned the process whose $STDIN is (or will be) this input spool file, as opposed to the *jobnum* of the process that streamed the job. Job numbers containing an apostrophe (J123) indicate that the spool file was imported by SPFXFER or RESTORE, or was recovered after a START NORECOVERY.
FILEDES	The formal or actual file designator for the spool file. Printing to a file equation such as FILE EPOCLONG;DEV=EPOC;ENV=LP88LONG.HPENV.SYS creates spool files whose formal designator is EPOCLONG.
DEV	The LDEV, device name, or device class name that is the destination of the spool file. LDEVs are intentionally displayed with leading zeroes to simulate a device name. When you specify LDEVs with SELEQ, you need not supply the leading zeroes.
PRI	The input or output priority of the spool file.
COPIES	The total number of copies of the spool file to be printed.
STATE	The current state of the spool file. READY and DELPND apply to input spool files as well as output spool files. ■ CREATE: An output spool file is being created; that is, an output spooled device has been opened and is being written to, generating an output spool file. When the device is closed, the spool file enters the READY state. ■ READY: An output spool file is ready to be printed, or an input spool file is ready to be accessed. ■ ACTIVE: An input spool file is active when it is being read from a STREAM file or a spooled device to disk. ■ OPEN: A JOB input spool file (the $STDIN for a batch job) is being accessed by the job's CI process or a DATA input spool file is being accessed by a process. ■ PRINT: An output spool file is being printed. If this command is entered while a trailer is being printed, you may observe two spool files in the PRINT state at the same time for the same device.

The native mode spooler does not open its own spool files. It calls a spool file management routine to select the next spool file and open it.

Is the following information still true?

In addition, there is a new NMS per-device configuration parameter that is currently not enabled because of a lack of a user interface for it. This allows you to specify that no banner (trailer and header) be printed between copies of the same spool file if copies are printed consecutively on the same device. The alternative is the default mode, namely that banners be printed between all copies of all spool files.

Because the output spooler does not select its own file and because it only prints one copy of a file at a time, it has no idea which file the file management routines will select for it. The spooler solves this problem by asking for its next spool file while the current spool file is open. It can then check to see if it should print a trailer and a header.

This means that the output spooler process can have two spool files open at the same time while it makes this test. But spool files opened by a spooler process are displayed in the `ACTIVE` state by `SHOWOUT` or the `PRINT` state by `LISTSPF`. As a result, it appears that the spooler is printing two files at the same time.

- `DEFER`: An output spool file is in the deferred state.
- `SPSAVE`: The `SPSAVE` option was specified when the spool file was created or at any time before it would have been deleted after its final copy was printed. That final copy has been printed, so the spool file is now in this state instead of being deleted.
- `PROBLM`: The target device of the spool file does not match any device name or device class on the system. This usually occurs because the spool file has been restored to a system that has a configuration different from the system from which the spool file was stored.
- `DELPND`: Either the spooler has printed the last copy of the output spool file and is waiting for one or more users to close the spool file before purging it, or someone has requested that the spool file be deleted and the file management routines are

waiting for the last FCLOSE of the spool file before purging it.

These commands place a spool file in the DELPND state:

- PURGE
- DELETESPOOLFILE
- SPOOLF *nnn*;DELETE
- STORE with the PURGE option

- XFER: The spool file has been selected for transportation from one node of a network to another. The XFER state is supported (in that it may be displayed, and used as a STATE in a selection equation), but is provided only for use as desired by third-party software providers. The spooler never places a file in this state nor uses the state as a basis for spooler actions.

RSPFN — The column under each letter R, S, P, F, and N, contains the respective letter as a flag indicating something about the spool file described in that row.

- R indicates a restartable spooled job file.
- S indicates that SPSAVE disposition has been specified for this spool file. The spool file are saved in the OUT.HPSPOOL group and account after the last copy is printed.
- P indicates that the spool file is private.
- F indicates that the spool file has a forms message associated with it and requires special forms on which to print. If a *formid* is present, its identity can be seen, using the ;DETAIL option, on the second line of the display for the given spool file.
- N indicates that the spool file is not complete because insufficient account-level, group-level or system disk space was available when the spool file was created or the system aborted while the spool file was being created.

OWNER — This is the fully qualified name of the creator of the spool file.

Display field and description:

The optional second line of the display has the following appearance:

```
FORMID   JOBNAME      COPSRM   SECTS   RECS   PAGES    DATE         TIME

         TESTJOB         1     250     500    ~9      12/20/88     8:39
```

`FORMID` An 8-character display, the first of which is a letter. If an *F* appears in the `RSPFN` column but this field is blank, it means that the file has a forms message but *formid* was not specified.

`JOBNAME` The job or session name of the user who created the spool file or, for a job input spool file, the name of the job that uses the input spool file as its `$STDIN` file..

`COPSRM` The number of copies of this file that remain to be printed, including any currently printing copy.

`SECTS` The number of sectors occupied by the spool file.

`RECS` The number of records in the spool file.

`PAGES` The number of physical pages in the spool file. This quantity is accurate only for CIPER protocol and 2680/88 page printers, and then only if the device has printed at least one complete copy. The device keeps track of the pages as they are printed and returns the correct count at the end of the copy. Until the actual count is known, an approximate count—calculated as *number_of_records* ÷ 60, and denoted by a leading tilde (~)—is displayed.

Note

For serial printers, even the count without the tilde is approximate because it is calculated as a best guess from the spool file data. It is not returned by the device because serial printers have no provisions for reporting this information.

`DATE` The date that the file first entered the `READY` state (*mm/dd/yy*).

`TIME` The time that the file first entered the `READY` state in 24-hour form (*hh:mm*).

The status display has the following format:

```
INPUT SPOOL FILES           OUTPUT SPOOL FILES
ACTIVE   = 1;               CREATE    = 2;       READY    = 3;
OPEN     = 2;               DEFER     = 1;       SELECTED = 4;
READY    = 3;               DELPND    = 0;       SPSAVE   = 1;
                            PRINT     = 1;       XFER     = 0;
                            PROBLM    = 0;


TOTAL IN FILES  =  6;       TOTAL OUTFILES    = 8;
      IN SECTORS = 144;           OUT SECTORS = 13090;

OUTFENCE  = 6
OUTFENCE  = 10 FOR LDEV  6
```

This display consists of three parts. The values in the first two parts represent only those spool files selected for display.

- The itemized count of spool files in each of the various states. They are shown in two groups, input spool files to the left of the display and output spool files to the right. Of these, only `SELECTED` is not a state. Instead, `SELECTED` shows the total count of spool files whose output priority is higher than the global outfence; that is, `SELECTED` displays the sum of printing files plus those `READY` files whose output priority is above the global outfence.
- The total number of input spool files, the sector count for input spool files, the total number of output spool files, and the sector count for output spool files.
- The global outfence and any device-specific outfences.

Use This command may be issued from a session, a job, a program, or in Break. It is breakable. Only files to which the user has access are displayed.

Examples

Following are some examples of the displays produced by LISTSPF. The first and third examples display all output spool files for the current *user.account* not using the console. The second example displays all spool files for the current *user.account* not using the console.

```
LISTSPF

SPOOLID  JOBNUM  FILEDES  PRI COPIES DEV          STATE RSPFN  OWNER

#0123    J12     SP        13      2 PP           PRINT    F   DEV.HPE
#0124    S14     LIST       9      1 00000012     READY    F   DEV.HPE
#0128    J144    $STDLIST   8      1 EPOC         READY        DEV.HPE
#01233   S1234   OUTLIST    0      1 FASTLP       DEFER        DEV.HPE

INPUT SPOOL FILES       OUTPUT SPOOL FILES
ACTIVE  = 0;            CREATE   = 0;        READY    = 2;
OPEN    = 0;            DEFER    = 1;        SELECTED = 3;
READY   = 0;            DELPND   = 0;        SPSAVE   = 0;
                        PRINT    = 1;        XFER     = 0;
                        PROBLM   = 0;

TOTAL IN FILES   = 0;   TOTAL OUTFILES    = 4;
      IN SECTORS = 0;         OUT SECTORS = 5964;

OUTFENCE  = 6
```

Single-Line Display (;DETAIL not specified)

```
LISTSPF @;DETAIL

SPOOLID JOBNUM   FILEDES PRI COPIES  DEV        STATE  RSPFN  OWNER
        FORMID   JOBNAME     COPSRM  SECTS      RECS   PAGES  DATE     TIME

#O123   J12      SP       13      2  PP         PRINT     F   DEV.HPE
                 TESTJOB          1  250        500      125  07/09/88 8:39

#O124   S14      LIST      9      1  00000012   READY     F   DEV.HPE
        PAYCHECK TESTJOB          1  250        500       ~9  12/20/88 8:39

#O128   J144     $STDLIST  8      5  EPOC       READY           DEV.HPE
                 LPJOB            3  250        127       21 12/20/88 22:19

#O1233  S1234    OUTLIST   0      1  FASTLP     DEFER DEV.HPE
                 TESTJOB          1  250        500       ~9  12/20/88 8:39

#I564   J164     $STDIN              00000010   READY          DEV.HPE
                 BATCHJOB            17         12             2/20/88  22:23


INPUT SPOOL FILES       OUTPUT SPOOL FILES
ACTIVE  = 0;            CREATE  = 0;                READY    = 2;
OPEN    = 0;            DEFER   = 1;                SELECTED = 3;
READY   = 1;            DELPND  = 0;                SPSAVE   = 0;
                        PRINT   = 1;                XFER     = 0;
                        PROBLM  = 0;

TOTAL IN FILES   = 1;   TOTAL OUT FILES    = 4;
      IN SECTORS = 17;        OUT SECTORS  = 1000;

OUTFENCE  = 6
```

Two-Line Display

```
LISTSPF;STATUS

INPUT SPOOL FILES       OUTPUT SPOOL FILES
ACTIVE  = 0;            CREATE  = 0;    READY    = 2;
OPEN    = 0;            DEFER   = 1;    SELECTED = 3;
READY   = 0;            DELPND  = 0;    SPSAVE   = 0;
                        PRINT   = 1;    XFER     = 0;
                        PROBLM  = 0;

TOTAL IN FILES   = 0;   TOTAL OUTFILES    = 4;
      IN SECTORS = 0 ;        OUT SECTORS = 1000;

OUTFENCE  = 6
```

Status Display

Related information

Commands	SPOOLF, SHOWIN, SHOWOUT, LISTFILE
Manuals	*MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)

OPENQ

Opens the spool queue for a specified logical device, device name, or all devices of a device class.

Syntax

```
OPENQ { ldev [;SHOW]
      { devclass [;SHOW]
      { devname [;SHOW]
      { @
```

Enhancements

devname — The device name of the spooled device. *devname* must begin with a letter and consist of 8 or fewer alphanumeric characters. Note that it is not possible to have a device class name and a device that are the same. If you enter an alphanumeric character string, the command searches the device class list first, and then the device name list.

SHOW — The SHOW parameter displays the current state (enabled or disabled) of the devices specified with the OPENQ command. If the spooling queues are globally disabled, the SHOW option displays this information.

@ — The @ parameter globally reenables all currently open spooling queues that were disabled because the system ran out of domain disk space, a file limit was encountered on the HPSPOOL account or its groups, or the SHUTQ @ command was entered.

If the spooling queues were disabled globally because the system is out of disk space or a file limit was encountered on the HPSPOOL account or its groups, the problem should be resolved before globally enabling spooling queues with the OPENQ @ command.

When you issue the OPENQ @ command, the following message is displayed on the system console:

```
ALL SPOOLING QUEUES CURRENTLY OPEN HAVE BEEN ENABLED.
```

When you issue the OPENQ *device* command where *device* is a logical device number, a device name class or a device name, while the spooling queues are globally disabled, you see the following message:

```
SPOOLING QUEUE OPENED FOR DEVICE device, BUT NOT IN
EFFECT SINCE THE SPOOLING QUEUES ARE GLOBALLY DISABLED.
(CIWARN 4625)
```

Refer to appendix B for more discussion on global enabling and disabling of spooling queues.

Use the @ option without any other parameter. The SHOW option entered with the @ option returns an error.

OUTFENCE

Defines the minimum priority an that output spool file needs in order to be printed.

Syntax

```
OUTFENCE outputpriority [;LDEV=dev]
[;DEV={dev | devclass | devname}]
```

Enhancements

dev — The logical device number of an output device.

devclass — The name of a class of devices that are to have their outfence value changed. *Devclass* must begin with a letter and consist of eight or fewer alphanumeric characters.

devname — *Devname* is the name of the device. It must begin with a letter and consist of eight or fewer alphanumeric characters. It is impossible to have a device class name and a device name that are the same. If you enter an alphanumeric string, the command searches the device class list first and then the device name list.

PURGE

Deletes a file from the system.

Syntax

PURGE *filereference*[,TEMP]

Enhancements

You may purge a nonprivate output spool file by entering PURGE *spool file* where *spool file* is the file name of the spool file. The PURGE command deletes the specified spool file, its spool file directory (SPFDIR) entry, and any checkpoint files that may exist for the spool file. The spool file does not print after it has been purged.

Note

The PURGE command cannot be used on a private spool file nor can it be used on any file to which it does not have exclusive access. For that reason, it cannot be used on a spool file in the PRINT state because the spool file has been opened by an output spooler process. Use the SPOOLF ... ;DELETE command instead.

RENAME

Changes identity (file name, lockword, and/or group name) of a disk file.

Syntax

`RENAME` *oldfilereference,newfilereference*[`,TEMP`]

Enhancements

You may rename spool files using the `RENAME` command if you have access to them. This is allowed only with spool files that are not linked to the spool file directory (`SPFDIR`).

SHOWDEV

Reports the status of input/output devices.

Syntax

```
SHOWDEV [ ldev | classname ] [ ;ACD ]
```

Parameters

ldev	Logical device number of device for which status information is to be displayed. This number is unique for each device. Default is that status information for all system devices on the system is displayed.
classname	Device class name of device(s) for which status information is to be displayed. This name may apply to several devices. Default is that status information for all devices on the system is displayed.
ACD	Keyword requesting display of access control definition (ACD)f or the device.

Operation notes

Displays the status information for all input/output devices on the system. The display appears in the following format:

```
SHOWDEV

LDEV AVAIL     OWNERSHIP    VOLID   DEN   ASSOCIATION

  1   DISC     43 FILES
  6   SPOOLED  SPOOLER OUT
  8   AVAIL
 20 A UNAVAIL  #S311:  7 FILES
```

The following items may appear in the listing, always displayed on the standard list device:

COLUMN	MEANING	
LDEV	Includes the logical device number and may include one of the following:	
	J	Accepts jobs
	D	Accepts data
	A	Accepts jobs and data
AVAIL	Lists the availability of devices and disks as follows:	
	AVAIL	The device is available as a real, nonshareable device.
	SPOOLED	The device is available for input or output spooling.

`UNAVAIL`	The device is not available; it is under the control of a job, session, or a system process, such as a spooler.
`DISC`	The device is a disk and is always available.
`DISC (RPS)`	The device is a CS-80 disk on which rotational position sensing (RPS) has been enabled.

OWNERSHIP — Includes device ownership and may include one of the following:

- SYS — Controlled by the system. If #*nnn* appears, it specifies the process identification number (PIN) of the controlling process (program).
- SPOOLER IN — Input spooling in effect, controlled by spooler.
- SPOOLER OUT — Output spooling in effect, controlled by spooler.
- #J*nnn* — Controlled by the indicated job.
- #S*nnn* — Controlled by the indicated session.
- *nn* FILES — Indicates number of files currently in use on a disk.
- DOWN — Device is offline, requested by system operator with the DOWN command.
- DP — Device is being taken offline (DOWN command operation pending).
- VOLID — The volume identification and may include one of the following:
  - IBM — The named magnetic tape volume that has a label written in the IBM format.
  - ANSI — The named magnetic tape volume that has a label.
  - NOLABEL — The named magnetic tape volume that has no label. Default.
- DEN — Density of the tape, which may include one of the following:
  - 6250 — Density of 6250 BPI (bytes-per-inch).
  - 1600 — Density of 1600 BPI, or the density of the tape is unrecognizable.

ASSOCIATION	Indicates the logical devices by device class that have been established by the user with the ASSOCIATE command.
ACD	Access Control Definition. May include any of the following information per username.acctname:

R	Read access.
W	Write access.
L	Lock access.
A	Append access.
X	Execute access.
None	No access.
RACD	Copy or read the ACD.

Use

This command may be issued from a session, a job, a program, or in Break. Pressing (Break) aborts the execution of this command.

Examples

To display the status of the device identified by logical device number 5, enter:

```
SHOWDEV 5

LDEV  AVAIL     OWNERSHIP     VOLID   DEN   ASSOCIATION
  5   SPOOLED   SPOOLER OUT
```

To display the status of all devices of the device class CARD, enter:

```
SHOWDEV CARD

LDEV  AVAIL     OWNERSHIP    VOLID   DEN   ASSOCIATION

  6 A AVAIL
```

SHUTQ

Closes the spool queue for the specified logical device, device name, or all devices of a device class.

Syntax

```
SHUTQ { ldev [;SHOW]
      { devclass [;SHOW]
      { devname [;SHOW]
      { @
```

Enhancements

devname — The device name of the device. Note that it is not possible to have a device class name and a device name that are the same. If you enter an alphanumeric character string, the command searches the device class list first, and then the device name list.

SHOW — The SHOW parameter displays the current queue state (enabled or disabled) of the devices specified with the SHUTQ command.

@ — The @ parameter globally disables all currently open spooling queues without closing the spooling queues. Thus, when the spooling queues are globally reenabled with the OPENQ @ command, all spooling queues that were opened before being globally disabled will again be open.

When you issue SHUTQ @, the following message is displayed on the system console:

```
ALL SPOOLING QUEUES HAVE BEEN GLOBALLY
DISABLED WITH THE 'SHUTQ @' COMMAND. USE
THE 'OPENQ @' COMMMAND TO GLOBALLY ENABLE
THE SPOOLING QUEUES.
```

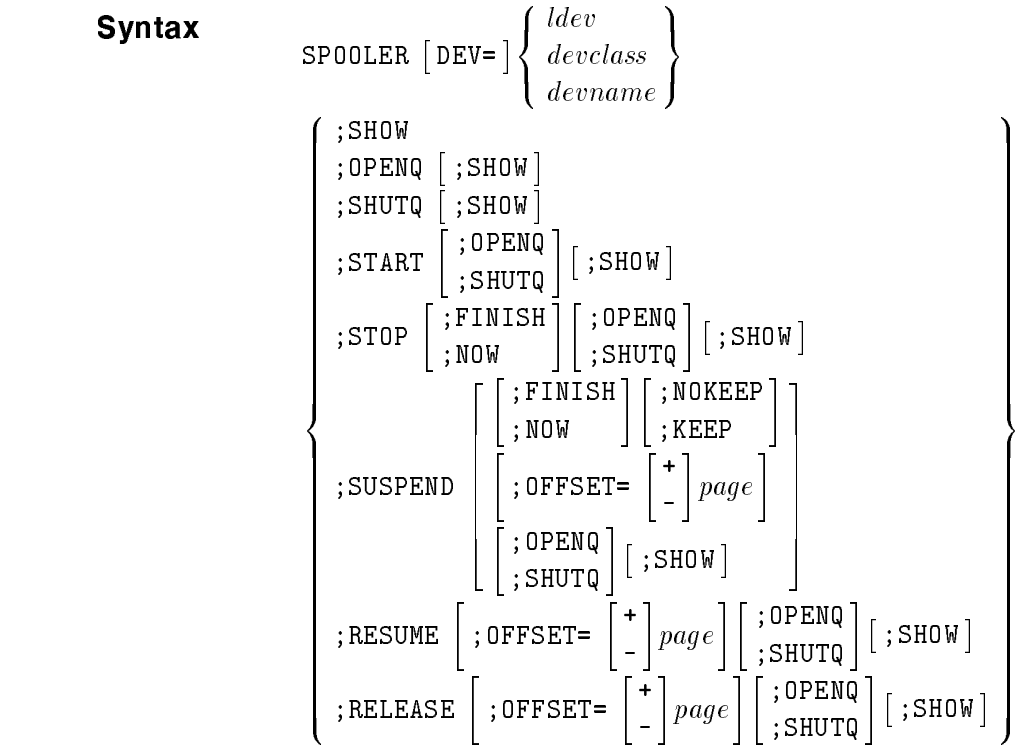
Refer to appendix B for more discussion on enabling and disabling of spooling queues.

Use the @ option without any other parameter. The SHOW option entered with the @ option returns an error.

SPOOLER

Controls spooler processes.

Syntax



Caution

When using network printers, avoid using the following commands to suspend the spooler in mid-file:

```
SPOOLER ... ; STOP
SPOOLER ... ; SUSPEND; OFFSET=anything
SPOOLER ... ; RESUME; OFFSET=anything
SPOOLER ... ; SUSPEND; NOKEEP
SPOOLER ... ; RELEASE
```

Many interfaces drop a network connection if the printer is ready to receive data but no data is being sent within a specific time period. The period is configurable at the printer or in the printer's TFTP file (specified in the bootptab entry), but many users simply use the factory default, which is 90 seconds.

The timer only runs when the printer is available but the host is not sending data, as is the case during a mid-file suspension. The timer does not run when the printer is unable to print, i.e., it has been taken offline, or places itself offline due to a paper out or toner low condition.

Parameters

ldev
The logical device number of the spooled device.

devclass
The device class name of the spooled devices. *devclass* must begin with a letter and consist of eight or fewer alphanumeric characters.

devname
The device name of the spooled device. The *devname* parameter must begin with a letter and consist of eight or fewer alphanumeric characters. Note that it is not possible to have a device class name and a device name that are the same. If you enter an alphanumeric character string, the command searches the device class list first, and then the device name list.

`START`
Output spoolers:

The `START` parameter creates and activates a new spooler process to own and manage the device and print spool files destined for it. If a class is specified, then a spooling process is created and activated for each device in the class.

If neither the `OPENQ` nor the `SHUTQ` option is specified, `OPENQ` is taken as the default.

Input spoolers:

The `START` parameter creates and activates a new spooler process to own and manage the device, to read data from it, and to create job or data input spool files for later processing by a CI (job) or user process (data). If a class is specified, then a spooling process is created and activated for each device in the class.

`STOP`
Output spoolers:

The `STOP` parameter terminates the spooling process associated with the specified device. If a class is specified, then spooling processes for all devices in the specified class are terminated.

A spooler in the active state first moves to the `STOP` pending state (shown as `*STOP` with the `SHOW` option) while it finishes its work on its current file (including any required trailer). When this is complete, or if the spooler was previously in the idle state, the spooler displays the following on the console (or the `$STDLIST` of an associated user) and terminates.

`Output spooler, LDEV #`*ldev*`: Stopped.`

You may determine the spooler state at any time by entering the following:

`SPOOLER` *ldev*`;SHOW`

or

SPOOLER *devclass*;SHOW

or

SPOOLER *devname*;SHOW

The STOP option is valid only if a spooler is in the ACTIVE, SUSPEND or IDLE state, or (if accelerating a previous STOP ;FINISH to STOP ;NOW) the STOP pending (*STOP) state.

If neither the NOW nor the FINISH option is specified, NOW is taken as the default.

If neither the OPENQ nor the SHUTQ option is specified, SHUTQ is taken as the default.

Note

Because of the large amount of data buffered in the file system and the device, an output device may continue to print, making it appear as if the STOP parameter has not had any effect. In reality, the spooler stops sending data to the device when the command is received but must wait until all buffered data has been printed before stopping. Depending on both the content of the data and the amount of buffering, this may require a significant part of a page or even several pages.

If the STOP is received while the spooler is printing a file, the page number of the last complete page that was printed is saved in the spool file's file label extension (FLABX). The next time that the file is selected for printing by any spooler, the output resumes at the page following the page number saved in the FLABX.

Input spoolers:

The STOP parameter terminates the spooling process associated with the specified device. If a class is specified, then spooling processes for all devices in the specified class are terminated. The spooler first moves to the STOP pending state (shown as *STOP with the SHOW option) while it finishes its work on its current file (closing and deleting it; rewinding the tape and placing it offline). When this is complete, the spooler displays the following message on the console (or the $STDLIST of an associated user) and terminates:

Input spooler, LDEV #*ldev*: Stopped.

You may determine the spooler state at any time by entering the following:

SPOOLER *ldev*;SHOW

The STOP option is valid only if a spooler is in the IDLE or ACTIVE state. Except for a short period during startup when it is in the START state, an input spooler is always in the IDLE or ACTIVE state.

The NOW, FINISH, OPENQ, and SHUTQ options are not applicable to an input spooler process and result in an error message if any is used.

SUSPEND

The SUSPEND option is valid only for output spooler processes. It suspends output to one or more spooled devices. The associated spooler processes remain alive, but inactive. A spooler in the ACTIVE state first moves to the SUSPEND pending state (shown as *SUSPEND with the SHOW option) while it finishes its work on its current file (including any required trailer). When this is complete, or if the spooler was previously in the IDLE state, the spooler displays the following on the console (or the $STDLIST of an associated user) and enters the SUSPEND state.

```
Output spooler, LDEV #ldev: Suspended.
```

If neither the NOW nor the FINISH option is specified, NOW is taken as the default.

If neither the KEEP nor the NOKEEP option is specified, KEEP is taken as the default.

If the OFFSET option is not specified, the spooler retains the present location in the output spool file. This is the default.

If you suspend the spooler with the KEEP option and want to force the spooler process to relinquish ownership of its spool file, do this:

```
SPOOLER 6;RELEASE
```

SUSPEND with certain other parameters presents special cases:

- SPOOLER *dev*;SUSPEND;NOW;KEEP with no ;OFFSET=

  (NOW and KEEP are defaults.)

  With no offset, the spooler suspends as soon as it processes the command. It suspends after processing the current spool file block and reads no more data from the spool file; nor does it flush existing data from the file system or device buffers.

  How you resume the spooling affects the subsequent data output.

  If you resume spooling by entering SPOOLER *dev*;RESUME, with no offset, the spooler continues from the point of interruption, and data in the buffer is printed in the normal course of operation. The output appears as though the spooler had never been suspended.

  If, instead, you resume spooling by entering

  ```
  SPOOLER dev;SUSPEND;NOW;KEEP
  ```

  with an offset, the spooler flushes all buffered data to paper *before* carrying out the offset request. As a result, you may see more output than you might expect from buffer flushing.

- SPOOLER *dev* ;SUSPEND with any other combination of different options (such as ;NOKEEP or ;OFFSET=).

  In such cases, the spooler knows that it will be resuming with a different file or at a different place in the same file. All buffered data is printed to paper before suspending. This may be a fair amount of buffered data.

- To avoid generating extra output, enter your offset when you suspend, instead of when you resume. For example:

  ```
  SPOOLER 16;SUSPEND;OFFSET=1
  :

  SPOOLER 16;RESUME
  ```

  Since you specify a page offset, all buffered data is printed to paper before the spooler suspends.

- Other considerations:

  If a spooler process is suspended in the middle of a spool file *and* the file is not retained by the

spooler, a page number is saved in the spool file's file label extension (FLABX). This page number is either the last complete page that was printed (if no OFFSET was specified) or one page prior to that specified by the final OFFSET applied to the file (with a lower limit of 0). The next time the file is selected for printing by any spooler, output resumes at the page following the page saved in the FLABX.

The SPOOLER *dev*;SUSPEND command is one example of a command that may interrupt a spooler process while it is printing a file. Other commands that may cause an interruption are:

```
=SHUTDOWN
 SPOOLER dev;SUSPEND;NOW              (SUSPENDSPOOL)
 SPOOLER dev;STOP;NOW                 (STOPSPOOL)
 SPOOLF  nnn;ALTER;DEV=dev            (ALTSPOOLFILE)
 SPOOLF  nnn;ALTER;DEFER              (ALTSPOOLFILE)
 SPOOLF  nnn;DELETE                   (DELETESPOOLFILE)
```

where *dev* is device class, the device name, or the logical device number and *nnn* is the spool file identification number (*spoolid*) or the set of files to be printed.

Any such command flushes all buffered data to the printer and updates the FLABX before terminating the spooler or releasing the file.

If *dev* is a device class, it is possible for the SPOOLER command to affect a spooler process that is printing a file and another process that is not. The same is true of the =SHUTDOWN command, which behaves like a SPOOLER ... ;STOP directed to all spooler processes. The considerations mentioned above apply only to a process that is printing a file when the command is issued. The command takes effect immediately on idle or previously suspended spooler processes.

Similarly, *nnn* may resolve to one or more spool files, some of which are printing—and some of which are not printing—when the command is issued. The considerations mentioned above apply only to the spool files that are printing. The command takes effect immediately on other spool files.

Beginning with release 2.2, the spooler displays the following for any of the above command situations:

```
Output spooler, LDEV #ldev: Received a
command while outputting a file.
```

This message is intended to reassure the operator that the spooler has accepted the command, since printing may continue for some time while buffered data is flushed.

`RESUME` The `RESUME` option resumes a suspended spooler process and is, therefore, valid only for output spoolers. The spooler must be in the `SUSPEND` state. If the spooler retains a spool file when it is suspended (meaning the `KEEP` option was specified or taken by default), and the spool file is not subsequently released, the `OFFSET` option is valid. If no offset is specified with either the earlier `SUSPEND` or the present `RESUME`, then output resumes where it left off. If an `OFFSET` is specified at either time (or both), the spooler resumes at the final location indicated by the offsets. If `OFFSET` is specified and the spooler does not have a retained file, a warning is generated and the spooler prints the next available spool file from the beginning.

`RELEASE` The `RELEASE` parameter directs a suspended output spooler to close (release) a spool file that it is currently retaining due to an earlier `SUSPEND ;KEEP` option. It is invalid and generates a warning if used in any other context. The `OFFSET` option may be used to change the resumption point of the file the next time it is selected for printing.

Note

When the file is released by the spooler, a page number is saved in the spool file's file label extension (FLABX). This page number is either the last complete page that was printed (if no `OFFSET` was specified) or one page prior to that specified by the final `OFFSET` applied to the file (with a lower limit of 0). The next time the file is selected for printing by any spooler, output resumes at the page following the page saved in the FLABX.

,

`FINISH` Directs the spooler to complete the currently active spool file and then suspend or stop. This option may be used only in conjunction with the `SUSPEND` or `STOP` options. If it is used in any other context, a warning is issued and the `FINISH` option is ignored. The `FINISH` parameter may not be used with either the `KEEP/NOKEEP` or `OFFSET` parameters.

Note The FINISH option is not valid for spooled input devices.

Either a STOP or SUSPEND that includes the FINISH option may be accelerated to a higher-priority command without waiting for the present spool file to finish printing. For example, SPOOLER ... ; SUSPEND; FINISH may be followed by:

```
SPOOLER...;SUSPEND;NOW
```

or

```
SPOOLER...;STOP;FINISH
```

or

```
SPOOLER...;STOP;NOW
```

Similarly, a SPOOLER ... ;STOP;FINISH may be accelerated to SPOOLER ... ;STOP;NOW. To go in the opposite direction is an error.

NOW — Directs the spooler to immediately stop the current output. This option may be used only in conjunction with the SUSPEND or STOP options. If it is used in any other context, a warning is issued. This is the default.

If NOW is used on the SUSPEND option with either the NOKEEP or OFFSET parameters, the spooler prints a trailer if required; otherwise output pauses and may be resumed later at the point of suspension.

Note The NOW option is not valid for spooled input devices.

KEEP — Valid only if all three of the following conditions are satisfied:

- KEEP is used as a parameter to the SUSPEND option.
- The spooler is actively processing a file or is suspending.
- The NOW parameter is also specified or taken by default.

Directs the device to retain ownership of the spool file that it is currently processing. This is the default.

If the OFFSET parameter is not specified (or this condition is taken by default), the spooler suspends after processing the current record.

NOKEEP — Valid only if all three of the following conditions are satisfied:

- NOKEEP is used as a parameter to the SUSPEND option.
- The spooler is actively processing a file or is suspending.
- The NOW parameter is also specified or taken by default.

Directs the spooler to close the spool file that it is currently processing. The spooler stops sending data after the current record, ejects a page, processes any specified OFFSET, saves the result of that processing (or the last completely printed page if no OFFSET was specified) in the FLABX (file label extension), prints a trailer with (INCOMPLETE) on it if trailers are enabled, and returns the file to the READY state. The next spooler that prints the file starts the first copy with the page following the page number saved in the FLABX and the file's header and trailer (if any) include (RESUMED) if printing starts anywhere but at the first page.

[+/-]*page* The *page* parameter may be used only in conjunction with the SUSPEND, RESUME, or RELEASE option. The *page* parameter must be an integer representing a physical page offset, either absolute or relative, within the file. Offsets are applied in the order in which they are entered, whether absolute or relative. If + is specified, the offset is adjusted forward relative to the current location by the number of pages specified. If − is specified, the adjustment is backward. If *page* is specified without + or −, then printing resumes at that page, absolute from the beginning of the file. No matter which combination of offsets are specified, the final location is limited by the bounds of the file.

Page Definition

A page is defined as follows:

- For CIPER protocol devices: a physical sheet.
- For the HP 2680 or HP 2688: a physical sheet (that may contain one or more logical pages).
- For serial printers: the OFFSET option (except for OFFSET=1 or OFFSET=0, the beginning of the file) is not reliable. No error or warning message is generated if it is used on such devices; however, results are unpredictable.

  This is because page numbers are accurate only for CIPER protocol devices and HP 2680 and HP 2688 page printers.

The spooler's serial printer storage manager makes an approximate guess as to the correct page. It is only a guess, however, based on

an extremely limited interpretation of the spool file by the storage manager, because a serial printer does not return page data to its storage manager.

The storage manager does not attempt to interpret the spool file data, looking for escape sequences that may advance paper, eject a page, or change the page length or line density. This would degrade performance to an unacceptable level. Instead, it checks the carriage-control character supplied as part of the user's `FWRITE` intrinsic call.

If that character is an ASCII "1" or an octal 300 (indicating skip to VFC channel 1, which by industry standard, is a page eject), `SPOOLER` notes that this type of page control is in use and assembles its own checkpoint based on the location of this record in the spool file. If a `RESUME` with `OFFSET` is later required, it counts these checkpoints to try to find the proper restarting point. The storage manager ignores any other carriage-control character.

The page-eject carriage-control is not required in user data, and many applications do not use it. In this case, the storage manager is forced to assume a static number of records (60) per page. Typically, this is the number of lines that fit on a standard 11-inch page at 6 lines per inch, allowing three lines of margin at the top and the bottom of the page. This is often a flawed assumption, as the following examples show:

- For many applications (for example, A4 paper, 8 lines per inch, and so on), 60 lines per page is the wrong value.
- Other applications are designed for specific forms and manage their own paper advancement. These applications may attach a carriage-control value to a record that causes paper to advance (say) five lines after printing a line of data. The storage manager counts this as one record.
- Control records (those that affect some aspect of printer operation but do not print anything) are included in the 60 record count.

The last two examples come about because the storage manager does not interpret the data in the spool file, as mentioned earlier, and so cannot detect these situations.

In summary, if the storage manager cannot interact with the device to determine page boundaries, it uses a carriage control "1" or %300, or 60 records per page to simulate checkpoints for `SPOOLER` *ldev*`;RESUME`. For the most consistent results with serial printers, you should always include an `OFFSET=1` parameter with the `SUSPEND` option. You can also include the parameter with a subsequent `RESUME` option, but this does not prevent another spooler process from printing the file from the "wrong" place in the meantime.

`SHOW` The `SHOW` parameter displays the status of the spooling process(es) associated with the device(s) specified. All other parameters on this command

are processed first, so the SHOW option reflects the updated state of the process(es) at the completion of the command executor. Please refer to the note following the example below.

OPENQ — The OPENQ option or parameter enables spooling for a specified logical device, device name, or all devices of a device class. This allows users to generate spool files on that device(s). Refer to the OPENQ command for more information.

OPENQ is the default value for the START option.

SHUTQ — The SHUTQ option or parameter disables spooling for a specified logical device, device name, or all devices of a device class. This prevents users from generating spool files on that device(s). Refer to the SHUTQ command for more information.

SHUTQ is the default value for the STOP option.

Note

At least one of the options must be specified for the SPOOLER command. SPOOLER DEV=PP is *not* a valid command; but SPOOLER DEV=PP;SHOW or SPOOLER DEV=PP; OPENQ; SHOW are valid commands.

Operation notes

The SPOOLER command allows you to start, stop, suspend, and resume spooler processes and to release files from the spooler process(es). Spooler processes come in two varieties: input spoolers and output spoolers.

An input spooler reads data from its device and uses that to create an input spool file. The data may consist of one or more batch jobs, data files, or any combination of the two. Input spool files are private files, meaning that they are only accessible to a user running in privileged mode. They are not printed, but are used strictly as input for other processes.

An output spooler processes output spool files—files that were created by a user accessing a spooled output device such as a printer or plotter. A spooled output device processes spool files first in order of priority and then by the time the spool file entered the READY state. Only files that have an output priority greater than the outfence are considered for output.

Note

Because this command may affect more than one process (if applied to all devices in a class), it is possible to get errors for some of those devices and not for others. For example, if class LP consists of LDEVs 6, 11, and 19, and LDEV 11 is already owned by a spooler process, the command SPOOLER LP;START creates and activates spooler processes for LDEVs 6 and 19, but also generates the message DEVICE 11 IS ALREADY SPOOLED.

Note

If you use any of the following commands to interrupt printing of a spool file (anywhere but at the end of the file) on a printer that does not support Page Level Recovery (PLR), the spooler displays a warning on your $STDLIST indicating it is initiating a recovery sequence. (Refer to appendix D for details on PLR.)

```
SPOOLER ... ; STOP
SPOOLER ... ; SUSPEND; OFFSET=anything
SPOOLER ... ; RESUME; OFFSET=anything
SPOOLER ... ; SUSPEND; NOKEEP
SPOOLER ... ; RELEASE
```

Examples

Following are examples of the OFFSET and SHOW options:

- A spooler is printing physical page 30 of its output, and the following sequence is entered:

  ```
  SPOOLER dev;SUSPEND;KEEP;OFFSET=-3
  SPOOLER dev;RESUME;OFFSET=-6
  ```

  Output resumes at page 21 (30−3−6=21).

- A spooler is again on page 30 when the following sequence is entered:

  ```
  SPOOLER dev;SUSPEND;KEEP;OFFSET=-15
  SPOOLER dev;RESUME;OFFSET=20
  ```

  Output resumes at (absolute) page 20.

- Under the same original conditions as the previous two examples:

  ```
  SPOOLER dev;SUSPEND;KEEP;OFFSET=20
  SPOOLER dev;RELEASE;OFFSET=-5
  ```

  The next time this copy is selected by a spooler, its output starts at page 15 (absolute page 20−5).

- To ensure that a file resumes at the beginning, enter:

  ```
  SPOOLER dev;SUSPEND;NOKEEP;OFFSET=1
  ```

An example of output using the SHOW option might be:

```
SPOOLER LP;SHOW

LDEV DEV      SPSTATE   QSTATE   OWNERSHIP      SPOOLID  JOB STEP (*)

 6   LDEV6    IDLE      OPENED   OUT SPOOLER
14   LDEV14   *SUSPEND  OPENED   OUT SPOOLER    #0237    DATA, WAIT FOR EOD
15   LDEV15   ACTIVE    OPENED   OUT SPOOLER    #0264    CONNECTING
19   LDEV19             OPENED   NO  SPOOLER
```

(*) *JOB STEP is for network printers only.*

Note

If the SHOW option is used, the display shows the current state of the selected spooler(s) *at the time that the command executor has completed processing the command*. This means that the selected spooler(s) may not actually be in the requested state, but in a pending state on the way to achieving the requested state. This is because it has not finished acting on the command and updating the process state before the SHOW option is performed. If this is so, an asterisk (*) precedes the process state on the SHOW display to denote that the state is pending. Please refer to LDEV 14 in the example display of the SHOW option above.

For network printers, any form of the SPOOLER command which includes the ;SHOW option includes a column labeled JOB STEP. If the spooler is printing a file, the JOB STEP column displays one of the following strings:

CONNECTING	*Trying to open a socket connection*
PRINTING HEADER	*Printing the header*
HDR, WAIT FOR EOD	*Waiting for the printer to finish the header*
PRINTING DATA	*Printing user data*
DATA, WAIT FOR EOD	*Waiting for the printer to finish user data*
PRINTING TRAILER	*Printing the trailer*
TRLR, WAIT FOR EOD	*Waiting for the printer to finish the trailer*
CLOSING CONN	*Trying to close the socket connection*
RECOVR BRKN CONN	*Trying to reopen a connection lost in midfile*

Note

The spooler skips the three WAIT FOR EOD job steps for printers which do not support Page Level Recovery (PLR). Such printers support only one-way communication (to the printer), and cannot report that they have finished printing anything.

Refer to the "Supported devices" section in chapter 2 for a list of printers which do and do not support PLR.

Use

This command may be issued from a session, a job, a program, or in Break. It is not breakable. It may be executed from the console or by a user to which the command has been allowed or associated.

Related information

Commands SPOOLF, LISTSPF, OPENQ, SHUTQ

Manuals *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)

SPOOLF

Allows a qualified user to alter, print, or delete output spool file(s). (**Native Mode**)

Syntax

SPOOLF has three possible execution branches. Which branch you choose depends upon whether your objective is to alter, print, ;delete.

Branch 1 (;ALTER)

```
SPOOLF { [IDNAME=] { spoolid                      }
                   { (spoolid[,spoolid] ... )     }
         [;ALTER] [ ;SELEQ= { [select-eq]     } ]
                            { ^indirect_file  }
         [ ;DEV= { ldev     } ]
                 { devclass }
                 { devname  }
         [;PRI=outpri] [;COPIES= numcopies]
         [;SPSAVE] [ ;DEFER   ] [;SHOW]
                   [ ;UNDEFER ]                        }
```

Branch 2 (;PRINT)

```
SPOOLF { [IDNAME=] { fileset                      }
                   { (fileset[,fileset] ... )     }
         ;PRINT [ ;DEV= { ldev     } ]
                        { devclass }
                        { devname  }
         [;PRI=outpri] [;COPIES= numcopies]
         [;SPSAVE] [ ;DEFER   ] [;SHOW]
                   [ ;UNDEFER ]                        }
```

Branch 3 (;DELETE)

```
SPOOLF { [IDNAME=] { spoolid                      }
                   { (spoolid[,spoolid] ... )     }
         ;DELETE [ ;SELEQ= { select-eq      } ]
                           { ^indirect_file }
         [;SHOW]                                       }
```

The ;ALTER keyword is optional. It is also the default for *all three* SPOOLF execution branches. If you do not specify ;ALTER, ;PRINT, or ;DELETE, SPOOLF accepts only those parameters and keywords associated with the first (;ALTER) branch.

Permitting ;ALTER to default has consequences: any attempt to specify parameters *not* belonging to the first execution branch fails.

If your objective is to alter, use the first execution branch and any of its parameters. If your objective is to print, use the second execution

branch and any of its parameters. If your objective is to delete, use the third execution branch and any of its parameters.

Caution

When using network printers, avoid using `SPOOLF ... ; DEFER` or `SPOOLF ... ; DEV=`*new LDEV* to suspend the spooler in mid-file. Many interfaces drop a network connection if the printer is ready to receive data but no data is being sent within a specific time period. The period is configurable at the printer or in the printer's TFTP file (specified in the bootptab entry), but many users simply use the factory default, which is 90 seconds.

The timer only runs when the printer is available but the host is not sending data, as is the case during a mid-file suspension. The timer does not run when the printer is unable to print, i.e., it has been taken offline, or places itself offline due to a paper out or toner low condition.

Parameters

spoolid

One or more spool file IDs: #I*nnn* for input spool files or #O*nnn* for output spool files. These IDs are assigned by the spooling subsystem at spool file creation time. The # is optional. So is the O if you are displaying output spool files; that is, if you specify neither [#]O nor [#]I, [#]O*nnn* is assumed. Do not attempt to specify a qualified file name. You must enter *spoolid*.

There is no default.

The symbol @ may be used to specify all spool files.
The symbol O@ may be used to specify all output spool files.
The symbol I@ may be used to specify all input spool files.
If @, O@, or I@ is specified, it must be the only value supplied. @, O@, and I@ are mutually exclusive.

Note

If you specify duplicate *spoolid*s, a warning message is displayed.

A console user or a user with SM or OP capability who specifies O@ acts on all output spool files on the system. A user with AM capability who specifies O@ acts on all output spool files created by users in the same account. All other users are limited to files they have created.

fileset

Specifies the set of files to be printed.

There is no default.

This positional parameter has this form:

filename[/*lockword*[.*groupname*[.*accountname*]]]

You may use wildcards. Files that are not spool files are ignored. An error is returned for each input spool file in the file set.

If the file name or set is not fully qualified, the default is the user's current logon group and account. In batch mode, if any file in the set has a lockword, it must be supplied with the command; therefore, the file cannot be part of a set that contains wildcards. This restriction does not apply in interactive mode because the system prompts the user for each required lockword. In any case, if the lockword is not correctly provided, the print option on that file fails with a warning message, and the command continues on the rest of the files, if any.

select-eq The selection equation is used as a filter on the set of spool files selected. Only spool files whose attributes satisfy all filter requirements are listed.

For example, you use the following command to delete all of the output spool files to which you have access and that have less than 100 pages from *user.acct*:

```
SPOOLF 0@;DELETE;SELEQ=[(OWNER=user.acct)AND(PAGES<100)]
```

Selection equations have the following format. In this display, when the expression is expanded, interpret the symbol ::= as "can be replaced by."

select-eq : := [*equation*]

Begin and end a selection equation with square brackets ([and]).

equation : :=
- *parm* { > | >= | < | <= | <> | = } *value*
- (*equation*)
- NOT *equation*
- *equation* { AND | OR } *equation*

Note The logical operator AND takes precedence over the logical operator OR. For example:

```
SPOOLF 0@;DELETE;SELEQ=[FILEDES=REPT OR OWNER=BOB.ACCTG AND PRI>8]
```

`[FILEDES=REPT OR OWNER=BOB.ACCTG AND PRI>8]` is the same as `[FILEDES=REPT OR (OWNER=BOB.ACCTG AND PRI>8)]`.

value ::= Appropriate values per data type.

parm ::= The parameter (*parm*) may be one of several attributes of the spool file to be altered or deleted, such as the *dev* parm, the `FILEDES` parm, and so on. The parm choices are described below.

Note For string types other than `DATE`, such as user name, only the relational operators "=" and "<>" apply. Using any others results in an error.

- *parm ::=* `DEV`: LDEV number, device name, or device class name. You may use wildcards for device name and device class name.
- *parm ::=* `FILEDES`: Formal or actual file designator for the spool file. For example, if you enter the file equation below and print to it, `EPOCLONG` is the spool file's `FILEDES`.

```
FILE EPOCLONG;DEV=EPOC;ENV=LPLONG.ENV.SYS
PRINT MYFILE,*EPOCLONG
```

  You may use wildcards.

  This keyword supports selection on the null string by entering `FILEDES= ""` (you may also use single quotes). You must include such a construct if you specifically want to select on such an attribute. Note that "" is not the same as " ". The blank is significant.
- *parm ::=* `SPOOLID`: Spool File identifier number in the format #O*nnn* or #I*nnn*.

  The # is optional; but if it is used, an O or I must also be used. If it is not used, the O is also optional for output spool files; that is 123 is the same as #O123. The valid range of *spoolid*s is $1 \leq nnn \leq 9{,}999{,}999$. (The commas are for clarity; do not enter any commas in the actual equation.)
- *parm ::=* `PAGES`: Number of pages in spool file (if known). Use a positive integer.

Note

This attribute does not apply to input spool files; therefore, any logical *condition* involving the attribute always returns FALSE when tested against an input spool file.

- *parm ::=* FORMID: Form name. You may use wildcards. (The *formid* is an ASCII string up to 8 characters, the first of which must be a letter.). Refer to the note accompanying the FILEDES and *pages* description.
- *parm ::=* STATE: READY, ACTIVE, OPEN, CREATE, PRINT, PROBLM, DELPND, SPSAVE, DEFER, XFER.
- *parm ::=* JOBNAME: Job or session name under which the spool file was created. The job name can consist of up to 8 alphanumeric characters, the first of which must be a letter.

  For a job input spool file, the JOBNAME shown is allocated to that job, *not* the job or session that streamed it.

  You may use wildcards.
- *parm ::=* DISP: Disposition can be SPSAVE or PURGE. Refer to the NOTE accompanying the PAGES description.
- *parm ::=*COPIES: Number of copies. Minimum is 1, maximum is 65,535. (The comma in 65,535 is for clarity; do not enter commas in the actual equation.)

Note

If printing has started and you wish to alter the number of copies to be printed, use the ALTSPOOLFILE command to make the change.

The SPOOLF command tracks the number of copies already printed. If you use SPOOLF to alter the number of copies to be printed, the spool file is automatically deleted if the new number requested is less than or equal to the number already printed.

The ALTSPOOLFILE command tracks the number of copies remaining to be printed. During printing, it does not delete the spool file if the number that you request is less than or equal to the number originally requested.

Refer to the *MPE/iX Commands Reference Manual* (32650-90003).

- *parm ::=* PRI: Output priority. Minimum is 0, maximum is 14. Refer to the note accompanying the PAGES description.
- *parm ::=* JOBNUM: Job or session number under which the spool file was created, for example: #S257, #J329, or J*n* (the "#" is optional). 1

$\leq n \leq$ 16,383. (The commas are for clarity; do not enter any commas in the actual equation.)

For a job input spool file, the JOBNUM shown is allocated to the job, *not* the job or session that streamed it.

You may use some wildcards; J@ accepts all jobs, S@ accepts all sessions. J'@ and S'@ are also allowed, The apostrophe (') indicates an imported spool file or a spool file recovered during START NORECOVERY.

- *parm ::=* RECS: Number of records in the spool file. A positive integer is expected.
- *parm ::=* OWNER: The user under which the spool file was created. The format of the *owner* is *user.account*. If the account is not specified, the user's current account is assumed. You may use wildcards.

  For a job input spool file, the OWNER is the user logon for the job, *not* the job or session that streamed it.
- *parm ::=* JOBABORT: Select based on whether this is the $STDLIST of a job that aborted when an error was encountered when no CONTINUE was in effect.

  Valid values are TRUE and FALSE. Only "=" and "<>" are allowed as relational operators. Refer to the note accompanying the PAGES description.
- *parm ::=* DATE: Creation date in the format *mm/dd/yy* or *mm/dd/year*. Note that the year can be in the form of *yy*, as in 10/10/88, or in the form of *year*, as in 10/10/1988; both are legal syntax for the *date* parameter.

indirect_file Specifies the name of a file containing the selection equation. It must be preceded by a caret (^). The selection equation contained in the file may not exceed 277 characters in length, including the brackets in which it must reside. There is no restriction on the indirect file code. If the record size exceeds 277, only 277 characters per record are read and a warning is issued. Backreferencing to a formal file designator is also allowed for an *indirect_file* name; that is, ^**filename* is also allowed. Any file is accepted as an *indirect_file*, unless the file system returns an error from FOPEN or FREAD.

There is no limit to the number of records in the *indirect_file*, only the total character count.

Records are processed as follows:

- Leading and trailing blanks are stripped.
- If the last nonblank character is an ampersand (&), it is also stripped; otherwise, one blank is added back to the end of the record as a delimiter.
- The character count of the record is added to that of the records processed previously. If the total character count exceeds 277, an error is returned. If the total is less than 277, the current record is appended to previous records.
- This process repeats until either 277 characters have been counted or the end-of-file is detected. Records terminating with or without ampersands may be mixed as desired in the indirect file.
- If the resulting string is $\leq$277 characters, it is parsed.
- If the parser detects a syntax error, or if any nonblank character follows the closing bracket (]) of the *select-eq*, an error is returned and the *select-eq* is not processed.

`ALTER` The `ALTER` option alters the characteristics of specified output spool files. Private output spool files may be altered in a limited fashion; only the keywords `PRI`, `DEFER`, and `UNDEFER` are allowed. A system manager (SM) user may also specify `DEV=`.

You cannot alter the attributes of spool files in the `SPSAVE` state.

Note

If you use the `DEFER` or `DEV` keyword on a spool file that is being printed, the spooler process printing the file is interrupted. The spooler process saves the page number of the last complete page that was printed in the spool files's file label extension (FLABX). The next time the file is selected for printing by any spooler, output resumes at the page saved in the FLABX.

If you use the `DEFER` or `DEV` keyword to interrupt printing of a spool file (anywhere but at the end of the file) on a printer that does not support Page Level Recovery (PLR), the spooler displays a warning on your $STDLIST indicating it is initiating a recovery sequence. (Refer to appendix D for details on PLR.)

Because of the large amount of data buffered in the file system and the device, an output device may continue to print, making it appear as if the `DEFER` or `DEV` keyword has not had any effect. In reality, the spooler stops sending data to the device when the command is received but must wait until all buffered data has been printed before releasing the spool file.

Depending on both the content of the data and the amount of buffering, this may require a significant part of a page or even several pages.

`PRINT` The `PRINT` option copies the specified file sets to the `HPSPOOL` account and links the new output spool files into the spool queues for printing. It is especially useful for generating more copies of a spool file in the `SPSAVE` state.

If the target device or class information exists in the file label extension, that device or class is used as the default.

The `DEV=` option may be used to override this default. If there is no target device in the file label extension or the device specified is not valid, the `DEV=` parameter must be specified or an error message results.

The default values of `PRI` (8) and `COPIES` (1) may also be overridden by user-specified parameters. You may specify `;DEFER` or `;UNDEFER` or `;SHOW` for the target spool file that you are creating.

Any changes that you apply through `;PRINT` apply *only* to the new copy of the spool file(s) that you are creating. The changes do not apply to your original spool file(s).

Note

The user of the `SPOOLF ... ;PRINT` command must have nonshareable device (`ND`) capability. Private files cannot be printed using the `PRINT` option.

`DELETE` The `DELETE` option purges all specified private or nonprivate spool files to which the user has access from the system.

If a spool file is not in use (opened by a user, or being printed or stored), it is purged immediately. If it is in use, it is placed in `DELPND` state. Any spooler process printing it is notified, and printing stops at that point. Each of these files is deleted when its last user closes it, except in the case of `STORE`, as described below.

Note

Because of the large amount of data buffered in the file system and the device, an output device may continue to print, making it appear as if the `DELETE` option has not had any effect. In reality, the spooler stops sending data to the device when the command is received but must wait until all buffered data has been printed before stopping.

Depending on both the content of the data and the amount of buffering, this may require a significant part of a page or even several pages.

The following command returns the spool file to its previous state from the DELPND state, if the command is issued before the file is actually deleted:

- SPOOLF *nnn*;ALTER

Interruptions to the spooling process are different, depending on whether the spool file was opened by a spooler or by a user process.

- Spool File opened by a spooler

  If a spooler is printing the spool file and has not yet closed the file, entering the command SPOOLF *nnn*;ALTER returns the file to the PRINT state. The spooler has already been interrupted and is in the process of cleaning up by printing all data and closing the file. The cleanup process is not interrupted nor is it reversed due to the SPOOLF *nnn*;ALTER command.

  Because the spooler has been interrupted while printing a spool file, it marks the spool file as incompletely printed when it closes it. The spool file is put into the READY state, where it can be selected for printing once again.

- Spool File opened by a user process

  A user process that has opened a spool file is not interrupted by the SPOOLF *nnn*;DELETE command nor by the subsequent SPOOLF *nnn*;ALTER command. When the user process eventually closes the spool file, the file disposition used during the close determines the fate of the spool file. The spool file returns to the state it was in before the user opened it, if it continues to exist.

STORE introduces a unique situation. If a spool file is being stored when anyone (including the output spooler upon completing the last copy of the file) requests that the file be deleted, the file is placed in DELPND, as described above, but it cannot be purged by closing the file because it is still in use by STORE. Even so, the STORE command does not purge the file when it finishes with it (unless STORE's user has specified the PURGE option), because it accesses the file at a level lower than that known by the NMS file management routines. Such a file remains in the DELPND state until one of the following occurs:

- Someone opens it and closes it (with PURGE, SPOOLF;DELETE, FCOPY, PRINT, or an editor).
- STORE completes and the PURGE option was selected.
- It is made ready by raising the number of copies such that after the SPOOLF ... ;ALTER completes, the number of copies to be printed exceeds the number already printed.

The DELETE option works on either DATA input spool files in the READY state, or all output spool files in the READY, PRINT, DEFER, SPSAVE, or PROBLM state. It does *not* work on job $STDIN files; use the ABORTJOB command for these files.

ldev Specifies the logical device number of the spool file's new destination device.

If the spool file is in the PRINT state, it is returned to the READY state. It may immediately enter the PRINT state on *ldev* if all requirements are met.

Note

Printing of a spool file is interrupted only if the newly specified target *ldev*, *devclass*, or *devname* is different from the previous target *ldev*, *devclass*, or *devname*.

devclass Specifies the new destination device class name for the spool file. If the spool file is in the PRINT state, it is returned to the READY state. It may immediately enter the PRINT state on a device in *devclass* if all requirements are met.

The *devclass* parameter must begin with a letter and consist of eight or fewer alphanumeric characters. Note that MPE/iX does not allow the same name to be configured as a device class name and a device name. efer to the note accompanying *ldev*.

devname Specifies the device name of the spool file's new destination device. If the spool file is in the PRINT state, it is returned to the READY state. It may immediately enter the PRINT state on *devname* if all requirements are met. Note that this occurs even if *devname* is the same as the device currently printing the file.

The *devname* parameter must begin with a letter and consist of eight or fewer alphanumeric characters. Note that MPE/iX does not allow the same name to be configured as a device class name and a device name. Refer to the NOTE accompanying *ldev*.

outpri Specifies the output priority of the designated spool files, where 0 is the lowest and 14 is the highest. Only an OP user or the console can specify an *outpri* of 14; other users are limited to 13.

The default is 8 with the PRINT option and no change for the ALTER option.

numcopies Specifies the number of copies of the designated spool files to be printed. The allowable range is 1 through 65,535. (The commas are for clarity; do not enter any commas in the actual command.)

The default is 1 for the PRINT option and no change for the ALTER option.

SPSAVE The SPSAVE option specifies that the selected spool files are not to be deleted after their last copy has printed. Instead they are retained in the HPSPOOL account in the SPSAVE state until deleted manually. Among other advantages, this option allows documents to be copied to user file space, to be reprinted without being reformatted, and so on.

Private spool files may not be saved.

Note

When a file enters the SPSAVE state, its priority is set to 8 and its number of copies is set to 1. This is so that it will have the proper defaults should it be printed later.

DEFER The DEFER option changes the spool file's state to DEFER. If it is currently in the PRINT state, its spooler is notified and printing stops at that point. (See the note about buffer retention under the DELETE option.) The spool file's priority remains unchanged. If this option is used with the PRINT option, the spool file is copied to OUT.HPSPOOL and linked to the spooling system, but the state of the spool file is DEFER. The spool file is not printed until a subsequent SPOOLF ... ;UNDEFER is entered.

Note

If the DEFER option is used on any file in the CREATE state (opened for original creation), the spool file only enters the DEFER state after it is completed (closed for the last time).

UNDEFER The UNDEFER option changes a spool file's state from DEFER to READY and causes a spooler to start printing it if the spool file is qualified for an idle printer to print. The spool file's priority remains unchanged.

SHOW — The SHOW option allows a user to display the results of the SPOOLF command. All other parameters are processed before the SHOW. Here is an example:

```
SPOOLF @@;SELEQ=[DEV=16];ALTER;PRI=8;SHOW

SPOOLID   JOBNUM FILEDES  PRI  COPIES   DEV        STATE  RSPFN     OWNER
#0414     J5     $STDLIST   8       1   00000016   READY            ALIX.MKT
#0416     J7     HOTSTUFF   8       2   00000016   READY            JACK.SALES
```

Operation notes

Input spool file attributes cannot be altered, but input spooled DATA files can be deleted. Private spool files may be altered in a limited fashion; only the keywords PRI, DEFER, UNDEFER, and DELETE are allowed. If the user has system manager capability, DEV= is also allowed.

The SPOOLF ... ;ALTER command can be used on problem state spool files to alter the device attribute so that the spool file becomes ready again. Most of the time, the spool file is in the problem state because the target device of the spool file is invalid.

You may wish to select for printing only those spool files that do require special forms, or only those that do not require special forms. One way to do this is to use the ;FORMID parameter. Use a file equation with ;FORMID to designate one device that requires special forms and use another file equation without the parameter to designate a printer that does not require special forms.

You may select files with no FORMID by specifying a null string (SELEQ=[FORMID=""]). The following example uses the LISTSPF command, but ;SELEQ works equally well with the SPOOLF command.

File equations such as the ones here are used to create the designations:

```
FILE NOFORMID;DEV=LP,2
FILE FORMID1;DEV=LP,2;FORMID=FORMID1;FORMS=Forms Message 1.
FILE FORMID2;DEV=LP,2;FORMID=FORMID2;FORMS=Forms Message 2.
```

The priorities are set low, to defer printing. This gives you time to use the LISTSPF command to examine the state of your output spool files.

Create two output files using each file equation.

```
listspf

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#0620      S327     NOFORMID   2      1 LP       READY        USER.ACCT
#0621      S327     NOFORMID   2      1 LP       READY        USER.ACCT
#0622      S327     FORMID1    2      1 LP       READY        USER.ACCT
#0623      S327     FORMID1    2      1 LP       READY        USER.ACCT
#0624      S327     FORMID2    2      1 LP       READY        USER.ACCT
#0625      S327     FORMID2    2      1 LP       READY        USER.ACCT

INPUT SPOOL FILES            OUTPUT SPOOL FILES
ACTIVE   = 0;                CREATE   = 0;              READY    = 6;
OPEN     = 0;                DEFER    = 0;              SELECTED = 0;
READY    = 0;                DELPND   = 0;              SPSAVE   = 0;
                             PRINT    = 0;              XFER     = 0;
                             PROBLM   = 0;

TOTAL IN FILES   = 0;        TOTAL OUT FILES   = 6;
      IN SECTORS = 0;              OUT SECTORS = 96;

OUTFENCE = 6
:
```

Qualify the LISTSPF command:

```
listspf;seleq=[formid=formid1]

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#0622      S327     FORMID1    2      1 LP       READY        USER.ACCT
#0623      S327     FORMID1    2      1 LP       READY        USER.ACCT

INPUT SPOOL FILES            OUTPUT SPOOL FILES
ACTIVE   = 0;                CREATE   = 0;              READY    = 2;
OPEN     = 0;                DEFER    = 0;              SELECTED = 0;
READY    = 0;                DELPND   = 0;              SPSAVE   = 0;
                             PRINT    = 0;              XFER     = 0;
                             PROBLM   = 0;

TOTAL IN FILES   = 0;        TOTAL OUT FILES   = 2;
      IN SECTORS = 0;              OUT SECTORS = 32;

OUTFENCE = 6
```

```
listspf;seleq=[formid=formid2]

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#0624      S327     FORMID2    2      1 LP       READY        USER.ACCT
#0625      S327     FORMID2    2      1 LP       READY        USER.ACCT

INPUT SPOOL FILES           OUTPUT SPOOL FILES
ACTIVE   = 0;               CREATE   = 0;              READY    = 2;
OPEN     = 0;               DEFER    = 0;              SELECTED = 0;
READY    = 0;               DELPND   = 0;              SPSAVE   = 0;
                            PRINT    = 0;              XFER     = 0;
                            PROBLM   = 0;

TOTAL IN FILES   = 0;       TOTAL OUT FILES   = 2;
      IN SECTORS = 0;             OUT SECTORS = 32;

OUTFENCE = 6

listspf;seleq=[formid=""]

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#0620      S327     NOFORMID   2      1 LP       READY        USER.ACCT
#0621      S327     NOFORMID   2      1 LP       READY        USER.ACCT

INPUT SPOOL FILES           OUTPUT SPOOL FILES
ACTIVE   = 0;               CREATE   = 0;              READY    = 2;
OPEN     = 0;               DEFER    = 0;              SELECTED = 0;
READY    = 0;               DELPND   = 0;              SPSAVE   = 0;
                            PRINT    = 0;              XFER     = 0;
                            PROBLM   = 0;

TOTAL IN FILES   = 0;       TOTAL OUT FILES   = 2;
      IN SECTORS = 0;             OUT SECTORS = 32;

OUTFENCE = 6
:
```

To print out one of the spool files that do not require special forms, do this:

```
SPOOLF 0621;ALTER;PRI=7
```

To print one of the spool files that do require special forms, do this:

```
SPOOLF 0624;ALTER;PRI=7
```

Use

This command may be issued from a session, a job, a program, or in Break, . The SPOOLF ... ;SHOW command is breakable. The actions, however, cannot be stopped by Break. It may be executed by any user. What files the user can access with the command depends on the user's capabilities.

If your need is only to list spool files, use the LISTSPF command.

SPOOLF @;SHOW, for example, must retrieve each SPFDIR entry *and* write it back. It locks the SPFDIR and JMAT tables for the duration of the command execution. On a system that has several thousand spool files, this can take tens of minutes.

During table locking, any of a number of vital user-initiated actions are prohibited, depending upon the status of the SPFDIR and JMAT tables. Among those that may be prohibited are:

- output spool file activity
- job logon
- spool file creation
- spooler processes attempting to obtain files to print
- session logon or logoff

As the number of spool files on the system increases, this locking period may become lengthy. In extreme cases, locking may continue for tens of minutes.

In addition, this use of SPOOLF defaults to ;ALTER and changes any spool file in the DELPND state back to its previous state, usually to READY or sometimes to PRINT.

LISTSPF also performs table-locking, but the duration of the locking is brief (less than one minute on a system that has several thousand spoolfiles) and does not become excessive. Nor does LISTSPF produce any subtle side-effects. Finally, it generates the same display as SPOOLF @;SHOW.

Related information

Commands	SPOOLER, LISTSPF, LISTFILE, ALTSPOOLFILE, DELETESPOOLFILE
Manuals	*MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)

5

Utilities

The Spoolfile Interface Facility (SPIFF)

The spoolfile interface facility (SPIFF) allows you to list, manipulate, and transfer spooled device files (spool files) that are created and maintained by MPE/iX. SPIFF is an MPE/iX replacement for the MPE CM SPOOK5 program.

SPIFF supports many (but not all) of the commands of the SPOOK5 program. Several existing features have been enhanced, and new features have been added. Wherever this has been done, the default is the closest approximation to SPOOK5 behavior. Where differences exist, they are described here.

SPIFF commands summary

These are the commands you can use with SPIFF:

ALTER	Alters the priority, number of copies, target device, or any combination of these attributes, of one spool file or many spool files.
APPEND	Appends all or part of one or many spool files to a new spool file. The first spool file processed by the command creates the new spool file. Subsequent spool files are appended to it.
BROWSE	Invokes the HPBROWSE utility, if it is available.
COPY	Copies all or part of one or many spool files to a new spool file.
DEBUG	Invokes the MPE/iX DEBUG facility if the SPIFF user has Privileged Mode (PM) capability.
EXIT	Terminates SPIFF, returning control to its parent process.
FIND	Locates a specified pattern in a specified range of the current spool file.
HELP	Displays information about SPIFF and its commands.
INPUT	Inputs one or more spool files from a tape created by SPOOK5 or SPFXFER.
LIST	Lists a line range of the currently TEXTed spool file to SPIFFOUT.
MODE	Controls the width and format of the displayed output of the LIST and FIND commands.
OUTPUT	Outputs one or more spool files to a tape in SPOOK5/SPFXFER format.

PURGE	Deletes one or more spool files from the system.
QUIT	Terminates SPIFF, returning control to its parent process.
SHOW	Displays information about one or more spool files.
STORE	Stores one or more files to tape using the MPE/iX STORE subsystem.
TEXT	Accesses an output spool file for use by the ALTER, APPEND, BROWSE, COPY, FIND, LIST, PURGE, and SHOW commands.
XPLAIN	Displays a summary of SPIFF commands.

Operation notes

To run SPIFF, enter this:

```
SPIFF
```

or this:

```
RUN SPIFF.PUB.SYS
```

SPIFF displays an identifying banner and its > prompt.

```
SPIFF A.00.00 (C) COPYRIGHT HEWLETT-PACKARD CO. 1992
>
```

SPIFF is now ready to accept any of the commands listed above.

All user input is case-insensitive (except for the default case of quoted search strings in the FIND command). In addition, non-SPIFF commands are passed to MPE/iX as entered—there is no up- or down-shifting.

Major differences between SPIFF and SPOOK5

SPIFF represents an enhancement of SPOOK5. SPIFF supports most, but not all, of the commands accepted by SPOOK5. In addition, SPIFF accepts new commands not available to SPOOK5.

New or changed features

- File equations for SPIFF's $STDINX.
- Opens $STDINX rather than $STDIN; entering a leading colon (:) does not cause end-of-file..
- MPE/iX command interface with or without a leading colon (:).
- The INFO string on the command line.
- Single letter command recognition of most commands.
- Native mode output display.
- Two new commands: STORE and BROWSE.
- Enhancements to the PURGE, MODE, FIND, and HELP commands.

File equations and formal file designators. SPIFF opens the formal file designator SPIFFIN as its $STDINX and the formal file designator SPIFFOUT as its $STDLIST. You may redirect these files as desired with a file equation. However the record width of any redirected SPIFFOUT should not be less than 80 bytes; otherwise displays and messages may generate an error when SPIFF directs them to SPIFFOUT.

Do not specify a REC= parameter in a file equation for any tape file. The wrong combination of values may be rejected with an error message.

End-of-file on $STDINX. Because SPIFF opens $STDINX (SPIFFIN), a colon in column one does not produce an end-of-file condition. Entering :EOD, however, does produce an EOF.

When SPIFF encounters a colon in column one, it strips the colon and passes the result to the CI for processing. If the result is a valid MPE/iX command, the command is executed; otherwise, an error message is returned.

```
SPIFF A.00.00 (C) COPYRIGHT HEWLETT-PACKARD CO. 1992
>:listf

FILENAME

MYFILE

>
```

The colon is not necessary with SPIFF. If it is not present, SPIFF processes a candidate command in the same way SPOOK5 did (local dictionary first, then the CI).

MPE/iX command interface. Any command not recognized by SPIFF, or any command preceded by a colon (:), is passed to MPE/iX's Command Interpreter through the HPCICOMMAND intrinsic, which will execute the command whether it is an MPE/iX command, UDC, command file, or program file. The RUN command is allowed by HPCICOMMAND.

Because some commands are not executed by HPCICOMMAND, SPIFF will not execute the following commands:

Commands Not Executed within SPIFF

ABORT	DO	HELLO	SETCATALOG
BYE	EOD	JOB	SHOWCATALOG
CHGROUP	EOJ	LISTREDO	REDO
DATA	EXIT	OPTION	RESUME

Command line INFO string. You may specify one command in the INFO string (for example, SPIFF;INFO="SHOW @.@"). SPIFF executes the specified command before displaying the first command prompt. One and only one command is allowed in the INFO string. The entire INFO string is parsed. If it contains a syntax error, an error message is displayed and the command is ignored.

You may slso specify the file name of a file containing SPIFF commands (... INFO=^*filename*) and SPIFF will read (FREAD) the the file, one record at a time, and execute the commands. You may specify any number of SPIFF commands, but each command must be a separate record.

Note

You cannot concatenate commands in the INFO= string or in the file of SPIFF commands.

Command recognition. With two exceptions, SPIFF recognizes the first letter or the full form of the command (for example, F or FIND). The two exceptions are APPEND and STORE. The abbreviations for these two commands require two letters (AP, ST) to distinguish them from the ALTER (A) and SHOW (S) abbreviations, respectively.

This differs from SPOOK5, which recognized commands by any leftmost subset of characters. For example, F, FI, FIN, and FIND all executed the FIND command in SPOOK5,.

Output display. SPIFF uses Native Mode Spooler (NMS) display routines and displays its output through the MPE/iX LISTSPF or the SPOOLF command. For example, the SHOW command executes through a LISTSPF display. SHOW;@ executes as LISTSPF;DETAIL.

Also, when you invoke MODE CONTROLS=ON, a subsequent LIST displays data using the MPE/iX PRINTSPF format.

New commands.

- STORE interfaces to the MPE/iX STORE facility.
- BROWSE invokes the HPBROWSE utility, if it has been installed on the system.

Enhancements to FIND.

- Case-insensitive FIND (F ^ "string"). You may set this as the default by MODE ^ = ON.
- Column-insensitive FIND without F@. You may set this as the default by using MODE @ = ON.
- FIND and display all occurrences of a string in the specified range. For FIND + "ERROR"' (the range defaults to ALL, as in SPOOK5). You may set this as the default by using MODE + = ON.
- Unprintable characters, such as ESC and Shift are converted to dots by default (for compatibility with SPOOK5). This can be overridden with the MODE command. Note that such characters can cause unpredictable display operation if sent to an output device in

their raw form. You may set non-conversion as the default by using MODE DOTS = OFF.

- Entering Ctrl Y during a search aborts the search or the display, depending upon which is occurring when you enter Ctrl Y.

Other enhancements

- Several other commands have been enhanced. For example, COPY and APPEND now support selection equations in determining their working fileset.
- The HELP command operation has been changed to resemble that of the MPE/iX help facility. Entering HELP (or its abbreviation, H) with no parameters displays an introductory screen and places you in an interactive mode until you exit the help facility. The X[PLAIN] command operation has not been changed.

Error and warning messages

SPIFF error and warning messages roughly parallel the corresponding SPOOK5 messages, although the text has been changed. In many cases the error or warning number has also changed.

Control-Y

Entering Ctrl Y affects the operation of the COPY, APPEND, FIND, HELP, XPLAIN, LIST, and PURGE commands (refer to the entries for each of these commands).

At other times (and with other commands), entering Ctrl Y has no effect.

Retained (SPOOK-like) features

The command line syntax of all retained (SPOOK5) commands has also been retained. However, any displays that result employ the NMS command formats.

Features not retained (from SPOOK)

- KILL command. SPOOK5's KILL command, used to terminate child SPOOK5 processes, is not supported, because SPIFF supports process creation with the MPE/iX's RUN command.
- LOCKED state. The Native Mode Spooler does not support a LOCKED state for spool files. As a result, TEXTing in a file, or OUTPUTting it to tape does not change its state. If you TEXT in a spool file in the READY state, then issue the SHOW * command, the file state continues to be displayed as READY.

Security

Console user

The SPIFF console user can access any spool file on the system, regardless of capabilities. This is consistent with the spool file access rights of MPE/iX commands.

By contrast, a SPOOK5 user at the console gained no additional spool file access rights by using the console. A user having neither SM nor AM capability could access only those files that she or he had created.

Other users

With SPIFF, a non-console user has spool file access rights that vary with the user's capabilities:

- An SM or OP user can access any spool file on the system.
- An AM user can access any spool file created by a user in the same account.
- A user with none of these capabilities can access only spool files she or he has created.

Private spool files

Access rights to private spool files are the same as for non-private spool files. However, you can modify only the priority attribute of such spool files. A user with SM capability can also modify the target device.

SPIFF commands

The SPIFF commands outlined in "SPIFF commands summary" are described in detail in the sections that follow.

ALTER

Alters the priority, number of copies, target device, or any combination of these attributes, of one spool file or of many spool files.

Syntax

```
> A[LTER] { spoolfileid [,spoolfileid [, ... ]] }
          { *                                   }
          { username[.acctname]                 }
          { seleq                               }

;{ option[,{ option}[, ... ]]}
```

where {*option*} is

```
{ D[EV]={ ldev    } }
{       { devclass} }
{       { devname } }
{ P[RI]=priority    }
{ C[OPIES]=numcopies}
```

Parameters

spoolfileid — An existing spoolid to which the user has access. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (#). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the # is used, the O must also be used. If the O is used without the #, the parameter is interpreted as a user name and will probably cause an error.

* — The current spool file—one that has been explicitly TEXTed in, or that is current because it is the last spool file processed by the COPY, APPEND, or BROWSE command. If this form is used without a current spool file, an error message is displayed.

username — The name of a user on the system. Full MPE wildcarding is supported. The SPIFF user must have access to files generated by *username*. Refer to "Security".

acctname — The name of an account on the system. Full MPE wildcarding is supported. Default: the logon account is assumed. The SPIFF user must have access to files generated by users in *acctname*. Refer to "Security".

seleq — A native mode spooler selection equation specifying the set of spool files to be altered. The equation must be enclosed in brackets, as it is in the following example:

```
ALTER [OWNER=MANAGER.SYS AND PRI<3];DEV=LP,PRI=8
```

This alters all spool files created by the user MANAGER.SYS that have priority less than 3.

If you choose this (*seleq*) form of file set selection, SPIFF inserts an OWNER=!HPUSER.!HPACCOUNT in its internal selection equation, *unless* you explicitly include your own OWNER definition. This prevents users with SM, OP, or AM capabilities from accidentally accessing files that they did not create.

Consult one of the following documents for more information about selection equation syntax and semantics:

- *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)
- MPE/iX online help facility

ldev — A logical device number of a printer whose spooling queues are open.

devclass — A device class containing at least one printer whose spooling queues are open.

devname — The device name of a printer whose spooling queues are open.

Note

It is not possible to have a device class name and a device name that are the same. If you enter an alphanumeric character string, the command searches the device class list first, and then the device name list.

priority — A number between 1 and 13.

numcopies — A number between 1 and 65535.

Operation Notes

The ALTER command (abbreviated A) changes the priority, number of copies, device specification, or any combination of these, of one spool file or a group of spool files. Spool files may be designated explicitly in a list (for example, #O12345, #O67890), by user and/or account, (for example, MYUSER.MYACCT) or by selection equation.

SPIFF executes the ALTER command by transforming its parameters into a form suitable for the MPE/iX SPOOLF command, then executing the SPOOLF command. Any SPOOLF execution errors are displayed as such.

The display following the ALTER command can be interrupted by entering CTRL Y. Any subsequent display is discarded. The ALTER command itself cannot be interrupted.

For any private spool file, only the PRI may be changed. A user with SM capability may also change the target DEV . The COPIES may not be changed.

Options may appear in any order. If a particular option appears more than once, the last such option is used. For example, `ALTER 15928;p=2,p=3`—the resulting priority is 3.

Example Assume that spoolid `#06490` exists and is accessible to you:

```
ALTER 6490;c=3,p=4

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#06490     S35      MYFILE     4      3 LP       READY        MYUSER.MYACCT
>
```

Suppose that spoolid `#06491` has been marked private:

```
ALTER 6491;p=4,c=2
SPOOLF (06491);ALTER;SHOW;COPIES=2;PRI=4
CANNOT ALTER COPIES ON SPOOLFILE "#06491". (CIWARN 4660)
SHOW


SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#06490     S35      MYFILE     4      3 LP       READY        MYUSER.MYACCT
#06491     S35      PRVAT      2      1 LP       READY    P   MYUSER.MYACCT

INPUT SPOOL FILES            OUTPUT SPOOL FILES
ACTIVE   = 0;                CREATE   = 0;              READY    = 2;
OPEN     = 0;                DEFER    = 0;              SELECTED = 0;
READY    = 0;                DELPND   = 0;              SPSAVE   = 0;
                             PRINT    = 0;              XFER     = 0;
                             PROBLM   = 0;

TOTAL IN FILES   = 0;        TOTAL OUT FILES   = 2;
      IN SECTORS = 0;              OUT SECTORS = 2128;

OUTFENCE = 6;
```

Note that the illegal attempt to modify the number of copies prevented the legal change of priority. Note, too, that the `SPOOLF` command resulting from the `SPIFF ALTER` command is also displayed.

APPEND

Appends all or part of one or many spool files to a new spool file. The first spool file processed by the command creates the new spool file. Subsequent spool files are appended to it.

Syntax

```
> AP[PEND]

[ [ spoolfileid [ ,spoolfileid [ , ... ] ] ;  ]                          ]
[ [ * ;                                        ] [ range [ ,filename ] ] ]
[ [ username [ .acctname ] ;                   ]                          ]
[ [ seleq ;                                    ]                          ]
[ END                                                                     ]
```

where *range* is

```
[ [ recnumber [ {+|-} offset ] ]   [ , count                           ] ]
[ [ * [ {+|-} offset ]         ]   [ / recnumber [ {+|-} offset ]      ] ]
[ [ FIRST [ {+|-} offset ]     ]   [ / * [ {+|-} offset ]              ] ]
[ [ LAST [ {+|-} offset ]      ]   [ / FIRST [ {+|-} offset ]          ] ]
[                                  [ / LAST [ {+|-} offset ]           ] ]
[ ALL                                                                    ]
```

Semantics

Range expressions are limited to:

(*first line number of file*) <= expression <= (*last line number of file*).

Lines are numbered from 0 to N-1. No error is generated for exceeding these limits; `SPIFF` simply limits the expression. For example, `FIRST-2` evaluates to `FIRST`.

The following situations, although syntactically valid, are semantic errors and are flagged as such:

- Using `APPEND END` when you have not opened an append *filename*.
- Specifying a first position range expression which evaluates to a greater line number than that of the second position range expression.

  Note that the expression, not its components, is tested. A range of `LAST/FIRST` is always an error, but a range of `LAST-20/FIRST+40` is valid for any file consisting of no more than 61 lines (numbered 0 to 60).
- Omitting the source file syntax (*spoolfileid*, *username*, etc.), or specifying * as the source file, unless you have a current spool file. A current spool file is one that has been explicitly `TEXT`ed in, or is current because it is the last spool file processed by the `COPY`, `APPEND` or `BROWSE` command.

If you have no current spool file, the source file specification (something other than *) is a required parameter.

Parameters

spoolfileid — An existing spoolid to which the user has access. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (`#`). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the `#` is used, the `O` must also be used. If the `O` is used without the `#`, the parameter is interpreted as a *username* and will probably cause an error.

username — The name of a user on the system. This parameter, when used with the optional *acctname*, specifies the set of spool files to append. Full MPE wildcarding is supported. The `SPIFF` user must have access to files generated by *username*. Refer to "Security".

acctname — The name of an account on the system. This parameter, when used with *username*, specifies the set of spool files to append. Full MPE wildcarding is supported. Default: the logon account is assumed. The `SPIFF` user must have access to files generated by users in *acctname*. Refer to "Security".

seleq — A native mode spooler selection equation specifying the set of spool files to append. The selection equation must be enclosed in brackets as in the following example which appends all spool files created by the user `MANAGER.SYS` with priority less than 3:

```
APPEND [OWNER=MANAGER.SYS AND PRI<3];ALL
```

If you choose this (*seleq*) form of file set selection, `SPIFF` inserts an `OWNER=!HPUSER.!HPACCOUNT` in its internal selection equation, *unless* you explicitly include your own `OWNER` definition. This prevents users with SM, OP, or AM capabilities from accidentally acessing files that they did not create.

Consult one of the following documents for more information about selection equation syntax and semantics:

- *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)
- MPE/iX online help facility

Refer to the discussion of selection equations for the `LISTSPF` or `SPOOLF` MPE/iX command.

range — The range of lines of the spool file(s) to append. By default, only the current record is appended.

Any line number specified that is outside the range of lines in the spool file will be handled as though FIRST or LAST was specified, as appropriate. If your range consists of two expressions, the first expression must evaluate to a number no larger than than the second.

filename — The name of the target file, the file being appended to. It must be a spooled file. You may specify *filename* with or without the backreferencing *, as long as the corresponding file equation already exists.

If you omit this parameter, SPIFF tries to open the target file using the formal file designator of the first source spool file.

The controlling file equation may have been canceled with a RESET command since the source spool file was created. In this case, the attributes of the source spool file are given to the target spool file.

Similarly, the file equation may have been redefined. In that case, the file equation will prevail, because it overrides any HPFOPEN parameters specified by SPIFF.

ALL — Specifies that all of the records in the spool file(s) should be appended. No other range element is allowed if this keyword is used.

FIRST — The first record in the spool file.

LAST — The last record in the spool file.

* —
- When used as a source file specification, * specifies the current spool file.
- When used as a *range* element, * specifies the current record in the spool file.

recnumber — An absolute record or line number of text in the spool file. Records are numbered starting with 0.

offset — A relative number of records before (-) or after (+) the specified record.

count — A numeric value, the number of lines to be appended, including the starting record.

END — Closes the current append file, terminating append access to it.

Operation Notes

The APPEND command (abbreviated AP) appends all or a range of one or more spool files to a new spooled devicefile (*filename* in the syntax above). Spool files may be designated explicitly in a list (#O12345, #O67890), by user and/or account (MYUSER.MYACCT), or by selection equation.

If you have a current spool file you may omit the source file specification and SPIFF will take its source from the current spool file. It is an error to omit the source file specification if you do not have a current spool file.

The last source file processed remains as the current spool file, regardless of any earlier current spool file. For example, if you have TEXTed in #O18450, but you then APPEND #O18451,#O18452;ALL, the current spool file at the end of the command is #O18452.

The target file must be a local spooled device file. An ordinary disc file, a spooled device file on a remote node, or a non-spooled device file (such as a tape drive) is not supported and, if specified, results in an error.

The FCOPY subsystem can be used to create such a target file, but this is not recommended: except for the remotely spooled device file, doing so deletes information from the target file, which is usually vital to printing data properly. Once this information has been deleted, it cannot be recovered.

Examples of such information include the following:

- The control field for HP2680 environment file records. Without this information, the data in the records appears as unintelligible random characters.
- Prespace/postspace information. Without it, all records are printed in postspace mode.
- Information that indicates whether or not the first byte of data in each record should be treated as carriage control.

A remotely spooled device file—one that exists on another system node—is not supported and, if specified, results in an empty spool file on the remote system and an error message. SPIFF cannot delete this empty spool file on the remote system.

Once a target device file has been opened, any spool files specified in this command or subsequent APPEND commands are appended to it until an APPEND END command is entered. At that time, the device file is closed and enters the READY state. The next APPEND command will open a new target device file.

Note

If you use the *seleq* or *username.acctname* form, and if this resolves to more than one file, the order in which these files are appended to the target file is determined by the underlying LISTSPF command generated by SPIFF.

In general, for a given device or class queue, the order of source spool files is determined first by output priority, then by the time they first entered the READY state. To ensure a specific order in the target file, enter an explicit list of spoolid's. These will be processed in left-to-right order.

Entering Ctrl Y during command execution stops the execution after the current record is transferred. The current target file remains open for possible use by subsequent APPEND commands. The current source file remains open as the current spool file.

When you interrupt an append operation with Ctrl Y, the identity of the last record transferred is usually not known. Therefore, you should regenerate the file or use it to create a new append file with the desired subset of records.

Examples

EXAMPLES OF *range*:

```
*/*+20
*-20/*
ALL
FIRST/LAST
*/LAST
LAST-100/LAST
FIRST,20
100/200
5
```

EXAMPLES OF COMMANDS:

```
FILE MYLP;DEV=LP
FILE MYPP;DEV=PP;ENV=MYENV

APPEND #06490;ALL
```

Creates a new spool file with attributes identical to #06490. The new spool file remains open until closed with APPEND END.

```
APPEND [OWNER=MYUSER.MYACCT];ALL,*MYPP
```

Creates a new spool file targeted to device class PP using environment file MYENV. Spool files belonging to MYUSER.MYACCT are appended to this environment information.

```
APPEND 101,102,103;ALL,*MYPP
```

This is the same as the previous example, except that an explicit list of *spoolfileid*'s has been used.

```
APPEND [OWNER=SOMEBODY.ELSE];ALL,*MYLP
The specified fileset contains no accessible spoolfiles. (SPERR 82)
```

The user does not have SM or OP capability or is not an AM user in the ELSE account.

```
APPEND 10000;ALL
One or more of the specified spoolfile(s) is invalid. (SPERR 44)
```

Spool file #010000 does not exist or is inaccessible to this user.

Assume that MYLP is a terminal for a session and that *spoolfileid* 101 was sent to it.

```
> :FILE MYLP=$STDLIST
> APPEND 101;ALL
The target of a COPY or APPEND command must be a local spooled devicefile (SPERR 124)
>
```

Redefining the attributes of MYLP makes it impossible to copy *spoolfileid* #0101 to MYLP.

```
>APPEND ALL, *MYPP
You have no current TEXT file (SPERR 81)
```

No source spool file is specified nor was one opened in an earlier command.

```
> T #0357
> FIND @ "header: Start APPEND at next line"
100 This is the end of the header: Start APPEND at next line
>APPEND */LAST, *MYPP
```

After LISTing or FINDing a line, the current record pointer is advanced to the next record. Here the APPEND will start with record number 101.

BROWSE

Invokes the HPBROWSE utility, if it is available.

Syntax

```
> B[ROWSE] [ spoolfileid ]
           [ *           ]
```

Semantics

The *spoolfileid* parameter is optional only if you have previously TEXTed in a spool file; otherwise it is a required parameter.

Parameters

spoolfileid — An existing spoolid to which the user has access. This specifies the source file to be browsed. To be interpreted as a spoolid, this parameter must begin with a number or with a pound sign (#). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the # is used, the O must also be used. If the O is used without the #, it is considered an invalid spool file id.

* — The current spool file—one that has been explicitly TEXTed in, or that is current because it is the last spool file processed by the COPY, APPEND, or BROWSE command. If this form is used without a current spool file, an error message is displayed.

Operation Notes

The BROWSE command invokes the HPBROWSE utility to provide a more powerful interface for viewing and pattern searching than is available in SPIFF. If *spoolfileid* is specified, any current spool file is closed and the specified spool file becomes the TEXT file. If * is specified, or if no *spoolfileid* is specified (the two forms are equivalent), the current spool file is used.

The spool file being browsed is left open as the current spool file when the utility terminates.

Note

Consult your HPBROWSE documentation for more information. Because HPBROWSE is not included with the Fundamental Operating System (FOS), it may not exist on your system. In this case, an error is generated and the command fails, although any spool file TEXTed in remains as the current spool file.

COPY

Copies all or part of one or many spool files to a new spool file.

Syntax

```
           [ spoolfileid [,spoolfileid [, ... ]];  ]
> C[OPY]   [ * ;                                   ]
           [ username[.acctname];                  ]
           [ seleq ;                               ]

[range[,filename]]
```

where *range* is

```
[ [ recnumber[{+|-}offset] ]  [ , count                    ] ]
[ [ * [{+|-}offset]        ]  [ / recnumber[{+|-}offset]   ] ]
[ [ FIRST [{+|-}offset]    ]  [ / * [{+|-}offset]          ] ]
[ [ LAST [{+|-}offset]     ]  [ / FIRST [{+|-}offset]      ] ]
[                             [ / LAST [{+|-}offset]       ] ]
[ ALL                                                        ]
```

Semantics

Range expressions are limited to:

(*first line number of file*) <= expression <= (*last line number of file*).

Lines are numbered from 0 to N-1. No error is generated for exceeding these limits; `SPIFF` simply limits the expression. For example, `FIRST-2` evaluates to `FIRST`.

The following situations, although syntactically valid, are semantic errors and are flagged as such:

- Specifying a first position range expression that evaluates to a greater line number than that of the second position range expression.

  The expression, not its components, is tested. A range of `LAST/FIRST` is always an error, but a range of `LAST-20/FIRST+40` is valid for any file consisting of no more than 61 lines (numbered 0 to 60).

- Omitting the source file syntax (*spoolfileid*, *username*, etc.) unless you have previously `TEXT`ed in a spool file. If you have no currently `TEXT`ed spool file, the source file specification is a required parameter.

Parameters

spoolfileid — An existing spoolid to which the user has access. This specifies the source of the data to be copied. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (`#`). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the `#` is used, the `O` must also be used. If the `O` is used without the `#`, the parameter is interpreted as a *username* and will probably cause an error.

username — The name of a user on the system. This parameter, when used with the optional *acctname*, specifies the set of spool files to copy. Full MPE wildcarding is supported. The `SPIFF` user must have access to files generated by *username*. Refer to "Security".

acctname — The name of an account on the system. This parameter, when used with the *username*, specifies the set of spool files to copy. If omitted, the logon account is assumed. Full MPE wildcarding is supported. The `SPIFF` user must have access to files generated by users in *acctname*. Refer to "Security".

seleq — A native mode spooler selection equation specifying the set of spool files to copy. The selection equation must be enclosed in brackets as in the following example that copies all spool files created by the user `MANAGER.SYS` with priority less than 3:

```
COPY [OWNER=MANAGER.SYS AND PRI<3];ALL
```

If you choose this (*seleq*) form of file set selection, `SPIFF` inserts an `OWNER=!HPUSER.!HPACCOUNT` in its internal selection equation, *unless* you explicitly include your own `OWNER` definition. This prevents users with SM, OP, or AM capabilities from accidentally accessing files that they did not create.

Consult one of the following documents for more information about selection equation syntax and semantics:

- *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)
- MPE/iX online help facility

range — The range of lines of the spool file(s) to copy. By default, only the current record is copied. Any line number specified that is outside the range of lines in the spool file will be handled as though `FIRST` or `LAST` was specified, as appropriate. If your range consists of two expressions, the first expression must evaluate to a number no larger the second.

filename	The name of the target file, the file being copied to. You may specify *filename* with or without the backreferencing *, as long as the corresponding file equation already exists. If you omit this parameter, SPIFF tries to open the target file using the formal file designator of the first source spool file. The controlling file equation may have been canceled with a RESET command since the source spool file was created. In this case, the attributes of the source spool file are given to the target spool file. Similarly, the file equation may have been redefined. If that case, the file equation will prevail, because it overrides any HPFOPEN parameters specified by SPIFF.
ALL	Specifies that all of the records in the spool file(s) should be copied. No other range element is allowed if this keyword is used.
FIRST	The first record in the spool file.
LAST	The last record in the spool file.
*	■ When used as a source file specification, * specifies the current spool file. ■ When used as a *range* element, * specifies the current record in the spool file.
recnumber	An absolute record or line number of text in the spool file. Records are numbered starting with 0.
offset	A relative number of records before (-) or after (+) the specified record.
count	A numeric value, the number of lines to be copied, including the starting record.

Operation Notes

The COPY command (abbreviated C) copies all or a range of one or more spool files to a new spooled device file (*filename*). The new spool file is closed at the end of the command and enters the READY state. Spool files may be designated explicitly in a list (for example, #012345, #067890), by user and/or account (MYUSER.MYACCT), or by selection equation.

If you have a current spool file, you may omit the source file specification and SPIFF will take its source from the current spool file. It is an error to omit the source file specification if you do not have a current spool file.

The last source file processed remains as the current spool file, regardless of any earlier current spool file. If you have TEXTed in

#018450, but you then COPY #018451,#018452;ALL, the current spool file at the end of the command is #018452.

The target file must be a local spooled device file. An ordinary disc file, a spooled device file on a remote node, or a non-spooled device file (such as a tape drive) is not supported and, if specified, results in an error.

The FCOPY subsystem can be used to create such a target file, but this is not recommended: except for the remotely spooled device file, doing so deletes information from the target file, which is usually vital to printing data properly. Once this information has been deleted, it cannot be recovered.

Examples of such information include:

- The control field for HP2680 environment file records. Without this information, the data in the records appears as unintelligible random characters.
- Prespace/postspace information. Without it, all records are printed in postspace mode.
- Information that indicates whether or not the first byte of data in each record should be treated as carriage control.

A remotely spooled device file—one that exists on another system node—is not supported and, if specified, results in an empty spool file on the remote system and an error message. SPIFF cannot delete this empty spool file on the remote system.

Entering Ctrl Y during command execution stops the execution after the current record is transferred. The current target file is closed. The current source file remains open as the current spool file.

When you interrupt a copy operation with Ctrl Y, the identity of the last record transferred is usually not known. Therefore, you should regenerate the file or use it to create a new target file with the desired subset of records.

Examples

EXAMPLES OF *range*:

```
*/*+20
*-20/*
ALL
FIRST/LAST
*/LAST
LAST-100/LAST
FIRST,20
100/200
5
```

EXAMPLES OF COMMANDS:

```
FILE MYLP;DEV=LP
FILE MYPP;DEV=PP;ENV=MYENV
```

```
COPY #O6490;ALL
```

Creates a new spool file with attributes identical to #O6490.

```
COPY [OWNER=MYUSER.MYACCT];ALL,*MYPP
```

Creates a new spool file targeted to device class PP using environment file MYENV. Spool Files belonging to MYUSER.MYACCT are copies to the target file following the environment information.

```
COPY 101,102,103;ALL,*MYPP
```

This is the same as the previous example, except that an explicit list of *spoolfileid*s has been used.

```
COPY [OWNER=SOMEBODY.ELSE];ALL,*MYLP
The specified fileset contains no accessible spoolfiles. (SPERR 82)
```

The user does not have SM or OP capability or is not an AM user in the ELSE account.

```
COPY 10000;ALL
One or more of the specified spoolfile(s) is invalid. (SPERR 44)
```

Spool file #O10000 does not exist or is inaccessible to this user.

Assume that MYLP is a terminal for a session and that *spoolfileid* 101 was sent to it.

```
> :FILE MYLP=$STDLIST
> COPY 101;ALL
The target of a COPY or APPEND command must be a local spooled devicefile (SPERR 124)
>
```

Redefining the attributes of MYLP makes it impossible to copy *spoolfileid* #O101 to MYLP.

```
>COPY ALL, *MYPP
You have no current TEXT file (SPERR 81)
```

No source spool file is specified nor was one opened in an earlier command.

DEBUG

Invokes the MPE/iX DEBUG facility if the SPIFF user has privileged mode (PM) capability.

Syntax

```
> D[EBUG]
```

Operation Notes

The DEBUG command (abbreviated D) allows a user with privileged mode capability to enter the MPE/iX DEBUG facility. This is seldom necessary in normal use.

Caution

The normal checks and limitations that apply to standard MPE/iX users are bypassed in privileged mode. It is possible for a privileged mode user of DEBUG to destroy file integrity, including the MPE/iX operating system software itself.

Additional Discussion

Refer to the detailed discussion of the DEBUG facility in the *System Debug Reference Manual* (32650-90013). Consult this document before attempting to use the DEBUG command.

EXIT

Terminates SPIFF, returning control to its parent process.

Syntax

```
> E[XIT]
```

Operation Notes

The EXIT command (abbreviated E) terminates the SPIFF process. Any current spool file is closed before SPIFF terminates. Control is returned to SPIFF's parent process. This is usually a command interpreter, but may be another application if that application supports creation of a child process. In particular, SPIFF may be run from within another copy of SPIFF, using the RUN command. EXITing the second copy of SPIFF returns control to the first instance of SPIFF.

The EXIT command and the QUIT command operate identically.

Note

It is not possible to suspend operation of a child SPIFF process, give control to a parent SPIFF process, and return to the child SPIFF process, as the SPOOK5 program permitted you to do. Each exit of a child process terminates that process and closes all open files associated with that process. Each child process starts afresh when created. As a result, the KILL command is not supported by SPIFF.

FIND

Locates a specified pattern in a specified range of the current spool file.

Syntax

```
            [ @ ]
> F[IND]    [ + ]  [ "string" ] [ ,range ]
            [ ^ ]
```

where *range* is

```
[ [ recnumber [ {+|-} offset ] ]   [ , count                              ] ]
[ [ * [ {+|-} offset ]         ]   [ / recnumber [ {+|-} offset ]         ] ]
[ [ FIRST [ {+|-} offset ]     ]   [ / * [ {+|-} offset ]                 ] ]
[ [ LAST [ {+|-} offset ]      ]   [ / FIRST [ {+|-} offset ]             ] ]
[                                  [ / LAST [ {+|-} offset ]              ] ]
[ ALL                                                                       ]
```

Semantics

The FIND command operates only on the currently TEXTed spool file. It is an error if used when there is no current spool file.

Range expressions are limited to:

(*first line number of file*) <= expression <= (*last line number of file*).

Lines are numbered from 0 to N-1. No error is generated for exceeding these limits; SPIFF simply limits the expression. For example, FIRST-2 evaluates to FIRST.

Specifying a first position range expression that evaluates to a greater line number than that of the second position range expression is an error.

Note

The expression, not its components, is tested. A range of LAST/FIRST is always an error, but a range of LAST-20/FIRST+40 is valid for any file consisting of no more than 61 lines (numbered 0 to 60).

Parameters

@ By default, FIND locates only those occurrences of "*string*" that begin at the first character of a line. Specification of @ causes FIND to locate occurrences of "*string*" without regard to their position in the line.

You can use the MODE @ = ... command to enable or disable the @ option. With the option enabled, it is not necessary to specify the @ as part of the FIND command; however, specifying it is not an error and has no effect on command operation.

`+`	By default, the `FIND` operation stops at the first occurrence of the pattern in the specified (or defaulted) line range. Specification of `+` causes `FIND` to locate all occurrences of the pattern in the range. You can use the `MODE + = ...` command to enable or disable the `+` option. With the option enabled, it is not necessary to specify the `+` as part of the `FIND` command; however, specifying it is not an error and has no effect on command operation.

Note

Do not confuse this use of `+` with that of the addition (positive offset) operator in a *range* expression.

`^`	By default, `FIND` is case sensitive. If you specify `"error"` as the search pattern, lines containing `Error` or `ERROR` will not be detected. Specifying `^` (caret) makes `FIND` case insensitive. You can use the `MODE ^ = ...` command to enable or disable the `^` option. With the option enabled, it is not necessary to specify the `^` as part of the `FIND` command; however, specifying it is not an error and has no effect on command operation.
"*string*"	The pattern to be located by `FIND`. The double quotes are required around the pattern. If you omit this parameter, `FIND` locates the first line in the specified or defaulted line range.
range	The subset of records of the spool file(s) to search. The default range is from the current record to the last one, that is `*/LAST`. Any line number specified that is outside the range of lines in the spool file will be handled as though `FIRST` or `LAST` was specified, as appropriate. If the range consists of two expressions, the first expression must evaluate to a number no larger than the second.
`ALL`	Specifies that all of the records in the spool file(s) should be searched. No other range element is allowed if this keyword is used.
`FIRST`	The first record in the spool file.
`LAST`	The last record in the spool file.
`*`	The current record in the spool file.
recnumber	An absolute record or line number of text in the spool file. Records are numbered starting with 0.
offset	A relative number of records before (-) or after (+) the specified record.

count A numeric value, the number of lines to be searched, including the starting record.

Operation Notes

The FIND command searches a range of records in the currently TEXTed spool file for a specified pattern string, displaying the first one found. A command option allows displaying all matches found in the range.

With no options specified, the FIND command searches for and displays the first case sensitive match of the specified string if the match begins in column 1 of the record. Any of the three options broaden the searching capabilities of the command. The options may be specified singly or in any combination. Thus @+^+@++@^ and ++++ are both valid. Refer to the "Syntax" description above.

Any or all of the three options may be enabled or disabled by using the appropriate option of the MODE command. If an option is enabled using MODE, it need not be specified in the FIND command. Enabling an option already enabled by the MODE command is not an error and has no effect on command operation.

The current record pointer is left at the record following the one that matches the specified pattern, *unless* the + option is specified or unless no match is found in the specified range. For either of these situations, the current record pointers is left at end-of-range + 1 (or at LAST, whichever is less).

Examples

EXAMPLES OF *range*:

```
*/*+20
*-20/*
ALL
FIRST/LAST
*/LAST
LAST-100/LAST
FIRST,20
100/200
5
```

EXAMPLES OF COMMANDS:

Command	Action
FIND +^@ "error", ALL"	Locates all records in the spool file containing the pattern "error", whether in upper or lower case (or a mixture thereof), anywhere in the line.
FIND ^@ "error", FIRST/100"	Restricts the search to the first occurrence of that same pattern, this time somewhere in the first 101 records.

FIND @ “error”, */LAST’’	Locates the first occurrence of the pattern whose case, upper or lower, exactly matches that specified, this time somewhere between the current record and the last.

HELP

Displays information about SPIFF and its commands.

Syntax

```
> H[ELP] [ FEATURES
         [ SUMMARY
         [ [ command [ ALL      ] ] ]
         [ [         [ PARMS    ] ] ]
         [ [         [ OPERATION] ] ]
         [ [         [ EXAMPLE  ] ] ]
```

Parameters

none — If you invoke HELP with no parameters, the Help Facility will be interactive until you enter E or EXIT. Refer to the "Operation Notes" section.

To enter the MPE/iX help facility, use :HELP. Be sure to insert a colon before the HELP command.

FEATURES — Displays a short description of new SPIFF features along with differences between SPIFF and SPOOK5.

SUMMARY — Displays all available commands and their syntax. This is the closest match to SPOOK5's HELP or XPLAIN commands.

command — A SPIFF or MPE/iX command. You may specify a command file to which you have read access or a UDC, as long as neither one contains OPTION NOHELP.

ALL — Displays all help text for the command. This consists of all three sections, PARMS, OPERATION and EXAMPLE.

PARMS — Displays the PARMS section of the command's help text.

OPERATION — Displays the OPERATION section of the command's help text.

EXAMPLE — Displays the EXAMPLE section of the command's help text.

Operation Notes

The HELP command (abbreviated H) invokes SPIFF's help facility. The facility has an interactive mode and a direct access mode. Both are described below. Either mode has a scrolling facility, that is, if a display requires more than 23 lines, you are prompted whether or not to continue the display.

- **Interactive access mode**

  If HELP is entered with no parameters, as in

  ```
  HELP
  ```

  it enters the interactive mode, displays a table of contents and double greater-than (>>) prompt, and awaits your input. Entering any topic in the table of contents produces a description of that topic. Entering any command name (except EXIT) produces the syntax for that command and a list of the keywords. Entering a keyword such as PARMS produces a listing of all the items for that keyword (all parameters in this case). You can append the keyword to the command name and get only the help text for that keyword of that command.

Note

Because E[XIT] exits the interactive mode, the only way to obtain a description of SPIFF's EXIT command is by entering H[ELP] EXIT.

- **Direct access mode**

  If HELP is entered with one or more parameters, it enters direct access mode. It displays the text specified by your parameter(s), then returns you to the SPIFF prompt. Depending on the parameter(s), you can display summary help text for all commands, a detailed description of a specified SPIFF or MPE/iX command, or a brief description of SPIFF features.

The Help response to keyboard input depends on what is happening at the time:

- At a scrolling continuation prompt—(xx/yy) Continue?—or while the output is in the process of being displayed, you are actually within the MPE/iX PRINT command. If the response is such that the PRINT command is terminated, you are returned to the appropriate SPIFF prompt (>> or >), depending on how you began the text display. Refer to the discussion of the PRINT command in the *MPE/iX Commands Reference Manual* (32650-90003) to determine its response to keyboard input.

  In addition to the responses described there, a CTRL Y response also terminates the PRINT command and returns you to an SPIFF prompt.

- At an interactive mode prompt (>>), entering E[XIT] or : terminates interactive mode, returning you to the SPIFF command prompt. CTRL Y has no effect. Pressing Return (Enter on some keyboards) causes SPIFF to display information up to the next keyword or command, or until 23 lines have been displayed, whichever comes first.

SPIFF HELP vs. MPE/iX HELP

Entering H[ELP] puts you in the interactive mode of SPIFF Help. Entering :HELP puts you in the interactive mode of of the MPE/iX help facility.

Entering H[ELP] followed by one or more parameters causes SPIFF to search its help file (SPHLP000.PUB.SYS) first. If it finds an entry for the first parameter, it displays appropriate text and returns to the SPIFF prompt. If it cannot find an entry corresponding to the first parameter, it passes the parameter list to the MPE/iX help facility. If an entry is found there, again, the appropriate text is displayed and you are returned to the SPIFF prompt. If the parameter is neither a SPIFF or an MPE/iX command, you get the MPE/iX message:

```
Can't find anything under this command or in table of contents.
```

If you enter :HELP followed by one or more parameters, SPIFF goes directly to the MPE/iX help facility (direct access mode) without checking SPIFF's HELP first.

Note

The full form of the :HELP command is required. The :H abbreviation will not work, because it is not a valid MPE/iX command.

INPUT

Inputs one or more spool files from a tape created by SPOOK5 or SPFXFER.

Syntax

```
> I[NPUT] { spoolfileid [,spoolfileid [, ... ]] } ; *tapefile
          { username[.acctname]                 }
```

Parameters

spoolfileid — An existing spoolid to which the user has access. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (#). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the # is used, the O must also be used. If the O is used without the #, the parameter is interpreted as a user name and will probably cause an error.

username — The name of a user on the system or @. This parameter, when used with the optional *acctname*, specifies the set of spool files to be input. The SPIFF user must have access to files generated by *username*. Refer to "Security".

acctname — The name of an account on the system or @. The parameter, when used with the *username*, specifies the set of spool files to be input. Default: the logon account is assumed. The SPIFF user must have access to files generated by users in *acctname*. Refer to "Security".

tapefile — The backreferenced name of a file equation that specifies a tape device or class.

Operation Notes

Syntax checking is performed by SPIFF, but the command is executed by running SPFXFER.PUB.SYS as a child process. Any errors generated by SPFXFER are reported.

SPIFF supports the INPUT command by invoking SPFXFER.PUB.SYS as a child process, passing the SPIFF command line to it in an internal temporary $STDIN file. This has several consequences:

- The tape interface format recognized by SPOOK5 and by SPFXFER requires a particular combination of values for the REC= parameter of a file equation. Do not specify a REC= parameter in your file equation. The wrong combination of values may cause SPFXFER to return an error without transferring your file(s). It is a good idea to limit your file equation to the form:

  ```
  FILE MYTAPE;DEV=TAPE
  ```

- SPFXFER does not support full wildcarding for either the *username* or the *acctname* parameters. The only wildcard permitted for either parameter is @.

- SPFXFER displays a prompt before reading the command from the temporary file. You cannot and need not respond to the prompt. If control does not return to you immediately, either SPFXFER is currently reading in the spool files as requested, or (more likely) SPFXFER is waiting for a console reply—indicating that the desired tape has been mounted, enabling it to access the tape drive specified by *tapefile
- SPFXFER requires a space between the I[NPUT] command and the source file specification. If you omit the space, as in > I@.@, SPFXFER reports an error.

Example Assume that spoolid #06490 exists on a tape device defined by FILE MYTAPE:

```
INPUT 6490; *MYTAPE
Invoking SPFXFER.PUB.SYS.  Ignore prompts until further notice. (SPMSG 185)

SPFXFER  A.03.00  (C) HEWLETT-PACKARD CO., 1989
>  Do not respond to this prompt

#FILE      ====>   #FILE     #JOB       DEV/CL             OWNER
#06490     ====>   #023195   #J'2       LP                MYUSER.MYACCT
>
SPIFF has regained control from SPFXFER.  (SPMSG 186)
```

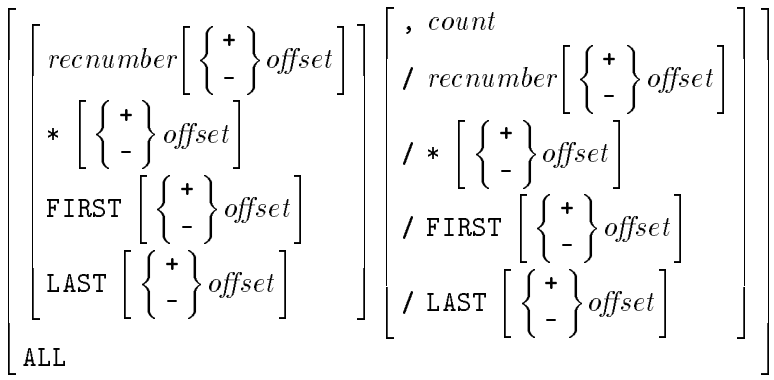
LIST

Lists a line range of the currently TEXTed spool file to $STDLIST.

Syntax

```
> L[IST] [ range ]
```

where *range* is



Semantics

The LIST command operates only on the currently TEXTed spool file. It is an error if used when there is no current spool file.

Range expressions are limited to:

(*first line number of file*) <= expression <= (*last line number of file*).

Lines are numbered from 0 to N-1. No error is generated for exceeding these limits; SPIFF simply limits the expression. For example, FIRST-2 evaluates to FIRST.

Specifying a first position range expression that evaluates to a greater line number than that of the second position range expression is an error.

Note

The expression, not its components, is tested. A range of LAST/FIRST is always an error, but a range of LAST-20/FIRST+40 is valid for any file consisting of no more than 61 lines (numbered 0 to 60).

Parameters

range — The range of lines of the spool file to list. By default (when the *range* parameter is omitted), only the current record is listed; this is equivalent to LIST *. Any line number specified that is outside the range of lines in the spool file will be handled as though FIRST or LAST was specified, as appropriate. If your range consists of two expressions, the first expression must evaluate to a number no larger than the second.

ALL — Specifies that all of the records in the spool file should be listed. No other range element is allowed if this keyword is used.

FIRST — The first record in the spool file.

LAST	The last record in the spool file.
*	The current record in the spool file.
recnumber	An absolute record or line number of text in the spool file. Records are numbered starting with 0.
offset	A relative number of records before (-) or after (+) the specified record.
count	A numeric value, the number of lines to be listed, including the starting record.

Operation Notes

The LIST command (abbreviated L) lists a range of records of the current spool file to $STDLIST. Omitting the *range* parameter is the same as specifying LIST *. In either case, the current record pointer is left at the record following the last record in *range*, or at LAST, whichever is less.

It is an error to use the LIST command if you do not have a current spool file.

Note

The ouput of the LIST command does not pause at nominal screen intervals the way that the output from HELP does. To list short intervals of the spool file, enter a command such as > LIST */*+10 as many times as you need to view the portion(s) of the file that you wish to see.

Examples

EXAMPLES OF *range*:

```
*/*+20
*-20/*
ALL
FIRST/LAST
*/LAST
LAST-100/LAST
FIRST,20
100/200
5
```

EXAMPLE OF A COMMAND:

```
LIST FIRST/10
  0
  1
  2 :JOB MYJOB,MYUSER.MYACCT.
  3 PRIORITY = DS; INPRI = 8; TIME = UNLIMITED SECONDS.
  4 JOB NUMBER = #J624.
  5 THU, DEC 19, 1991, 10:24 AM.
  6 HP3000  RELEASE: B.30.00   USER VERSION: B.30.00
  7 MPE XL  HP31900 B.08.14  Copyright Hewlett-Packard 1987.
  8 All Rights Reserved.
  9 STREAMED BY MYUSER.MYACCT (#S385) ON LDEV# 43
 10    STREAM DATE:   THU, DEC 19, 1991, 10:24 AM
```

MODE

Controls the width and format of the displayed output of the LIST and FIND commands.

Syntax

```
> M[ODE] { option [,option [, ... ]] }
```

where {*option*} is

```
{ W[IDTH]= { [+] nnn }
           { [-]     }
           { OFF     }
  C[ONTROLS]= { ON  }
              { OFF }
  @ = { ON  }
      { OFF }
  + = { ON  }
      { OFF }
  ^ = { ON  }
      { OFF }
  D[OTS]= { ON  }
          { OFF }
}
```

Parameters

nnn

An integer indicating the maximum number of characters to be displayed per record by LIST and FIND. A positive value indicates display width in MPE/iX halfwords (one halfword = 2 bytes), while a negative value indicates width in bytes. The range of *nnn* is -32767 to +32767; however, operating system limits are reached well before these limits.

WIDTH

WIDTH=OFF (default) causes SPIFF to display records in their entirety, except as limited by the width of the $STDLIST output device.

WIDTH=[+/-] *nnn* limits the display width of records displayed by the LIST and FIND commands. A negative number indicates a width in bytes, and a positive number indicates a width in 16-bit words (2 * *nn* bytes).

WIDTH=[+/-] *nnn* defines the maximum width of the displayed record. The record includes a space for at least three digits of line number and a separating blank. Beyond line 999, additional digits are added to the line number as required, and the data is moved a corresponding number of bytes to the right, truncating the same number of additional bytes from the right end of the record.

With WIDTH=80 in effect, for example, as many as 76 characters of data per record are displayed for records between 0 and 999. For records 1000/9999,

as many as 75 characters of data per record are displayed.

Only the display width is affected. The FIND command can successfully match a pattern even if the pattern lies partly or entirely beyond that part of the record that is displayed.

The + and - are optional. If either is used, there must be no space between it and the number that follows.

CONTROLS

CONTROLS=OFF (default) disables formatted display of I/O control information.

CONTROLS=ON enables formatted display of I/O control information. When CONTROLS=ON is in effect, the display is the same as that used by the PRINTSPF utility.

@

@=OFF (default) causes the FIND command to match its specified pattern only if the matching pattern begins at the first character of a record.

@=ON allows FIND to match its pattern anywhere in a record.

+

+=OFF (default) causes the FIND command to stop at the first matched pattern in its specified (or defaulted) line range.

+=ON causses FIND to locate all matching patterns in the range.

^

^=OFF (default) causes the FIND command to match its pattern only if the individual bytes in the candidate record match those of the pattern in a case-sensitive fashion. For example, "A" matches "A", but not "a".

^=ON results in a case-insensitive search.

DOTS

DOTS=ON (default) results in a translation of all non-printing characters (such as Ctrl N and ESC) to dots before being displayed by the LIST or FIND command.

DOTS=OFF disables such translation. Characters are sent to the display device exactly as they occur in the file.

Caution

Many display devices and communication links respond to escape sequences and other non-printing characters. Similar sequences may occur randomly in binary data. You should not set `DOTS=OFF` if your communication link may be adversely affected. You should set your terminal to "Display Functions" before displaying any unknown data. This mode is intended primarily for checking or debugging output data.

Operation Notes

The `MODE` command (abbreviated `M`) controls the width and format of `LIST` and `FIND` spool file record displays. Settings assigned with this command remain in effect until changed by another `MODE` command.

`MODE` allows you to concatenate as many options as you wish, as long as each is separated from the next by a comma. In case of duplication or conflict, the rightmost option is used.

Examples

Assume that the spool file used in the example for the LIST command is also the currently TEXTed spool file here. Here is the same fragment of display output:

```
LIST FIRST/10
  0
  1
  2 :JOB MYJOB,MYUSER.MYACCT.
  3 PRIORITY = DS; INPRI = 8; TIME = UNLIMITED SECONDS.
  4 JOB NUMBER = #J624.
  5 THU, DEC 19, 1991, 10:24 AM.
  6 HP3000  RELEASE: B.30.00   USER VERSION: B.30.00
  7 MPE XL  HP31900 B.08.14  Copyright Hewlett-Packard 1987.
  8 All Rights Reserved.
  9 STREAMED BY MYUSER.MYACCT (#S385) ON LDEV# 43
 10    STREAM DATE:   THU, DEC 19, 1991, 10:24 AM
> FIND + "STREAM", ALL
  9 STREAMED BY MYUSER.MYACCT (#S385) ON LDEV# 43
```

The "STREAM" in line 10 is not matched because no @ is in effect.

```
> MODE @=ON, += ON
> FIND + "STREAM", ALL
  9 STREAMED BY MYUSER.MYACCT (#S385) ON LDEV# 43
 10    STREAM DATE:   THU, DEC 19, 1991, 10:24 AM
```

The + is redundant here, but is not an error.

```
> FIND "copyright", ALL
```

The "Copyright" in line 7 is not matched because this is a case-sensitive search. With the appropriate MODE setting:

```
> MODE ^=ON
> FIND "copyright", ALL
  7 MPE XL  HP31900 B.08.14  Copyright Hewlett-Packard 1987.
```

```
> MODE WIDTH=-40
> LIST 0/10

  0
  1
  2 :JOB MYJOB,MYUSER.MYACCT.
  3 PRIORITY = DS; INPRI = 8; TIME = UNL
  4 JOB NUMBER = #J624.
  5 THU, DEC 19, 1991, 10:24 AM.
  6 HP3000  RELEASE: B.30.00   USER VERS
  7 MPE XL  HP31900 B.08.14  Copyright H
  8 All Rights Reserved.
  9 STREAMED BY MYUSER.MYACCT (#S385) ON
 10    STREAM DATE:   THU, DEC 19, 1991,
```

```
MODE WIDTH=-60,CONTROLS=ON
LIST 0/10
  0 OP P1=$0000 P2=$0000
  1 WR P1=$0001 P2=$0000 CC=%061
  2 WR P1=$0001 P2=$0000 CC=%000 BUF/#  25= :JOB MYJOB,MYUSE
  3 WR P1=$0001 P2=$0000 CC=%000 BUF/#  51= PRIORITY = DS; I
  4 WR P1=$0001 P2=$0000 CC=%000 BUF/#  19= JOB NUMBER = #J6
  5 WR P1=$0001 P2=$0000 CC=%000 BUF/#  28= THU, DEC 19, 199
  6 WR P1=$0001 P2=$0000 CC=%000 BUF/#  48= HP3000  RELEASE:
  7 WR P1=$0001 P2=$0000 CC=%000 BUF/#  56= MPE XL  HP31900
  8 WR P1=$0001 P2=$0000 CC=%000 BUF/#  20= All Rights Reser
  9 WR P1=$0001 P2=$0000 CC=%000 BUF/#  40= STREAMED BY MYUS
 10 WR P1=$0001 P2=$0000 CC=%000 BUF/#  45=    STREAM DATE:
```

OUTPUT

Outputs one or more spool files to a tape in SPOOK5/SPFXFER format.

Syntax

```
> O[UTPUT] { spoolfileid [,spoolfileid [, ... ]] } ; *tapefile
           { username[.acctname]                 }
[;PURGE]
```

Parameters

spoolfileid — An existing spoolid to which the user has access. Specifies the source of the data to be output to tape. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (#). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the # is used, the O must also be used. If the O is used without the #, the parameter is interpreted as a user name and will probably cause an error.

username — The name of a user on the system or @. This parameter, when used with the optional *acctname*, specifies the set of spool files to be output. Full wildcarding is not supported by SPFXFER. The SPIFF user must have access to files generated by *username*. Refer to "Security".

acctname — The name of an account on the system or @. Default: the logon account is assumed. This parameter, when used with the *username*, specifies the set of spool files to be output. Full wildcarding is not supported by SPFXFER. The SPIFF user must have access to files generated by users in *acctname*. Refer to "Security".

tapefile — The backreferenced name of a file (FILE) equation that specifies a tape device or class.

PURGE — If specified, spool files are deleted from the system after being output to tape.

Operation Notes

Syntax checking is performed by SPIFF, but the command is executed by running SPFXFER.PUB.SYS as a child process. Any errors generated by SPFXFER are reported.

SPIFF supports the OUTPUT command by invoking SPFXFER.PUB.SYS as a child process, passing the SPIFF command line to it in an internal termporary $STDIN file. This has several consequences:

- The tape interface format recognized by SPOOK5 and by SPFXFER requires a particular combination of values for the REC= parameter of a file equation. Do not specify a REC= paramater in your file equation. The wrong combination of values may cause SPFXFER to return an error without transferring your file(s).

  It is a good idea to limit your file equation to the form:

```
FILE MYTAPE;DEV=TAPE
```

- SPFXFER does not support full wildcarding for either the *username* or the *acctname* parameters. The only wildcard permitted for either parameter is @.
- SPFXFER requires a space between the O[UTPUT] command and the source file specification. If you omit the space, as in > O@.@, SPFXFER reports an error.
- SPFXFER displays a prompt before reading the command from the temporary file. You cannot and need not respond to the prompt. If control does not return to you immediately, either SPFXFER is currently outputting the spool files to tape as requested, or (more likely) SPFXFER is waiting for a console reply—indicating that the desired tape has been mounted, enabling it to access the tape drive specified by *tapefile.

Example Assuming that spoolid #06490 exists, and that the user has access to it:

```
OUTPUT 6490; *MYTAPE
Invoking SPFXFER.PUB.SYS.  Ignore prompts until further notice. (SPMSG 185)

SPFXFER  A.03.00  (C) HEWLETT-PACKARD CO., 1989
>  Do not respond to this prompt

#FILE         #JOB      DEV/CL         SECTORS          OWNER
#06490        #S2       LP             16              MYUSER.MYACCT
>
SPIFF has regained control from SPFXFER.  (SPMSG 186)
```

PURGE

Deletes one or more spool files from the system.

Syntax

```
              { spoolfileid [,spoolfileid [, ... ]]          }
> P[URGE]     { *                                            }
              { { username[.acctname] } [;{ ASK }]           }
              { { seleq               }   { YES }            }
```

Parameters

spoolfileid An existing spoolid to which the user has access. This parameter must begin with a number or with a pound sign (#). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the # is used, the O must also be used. If the O is used without the #, the parameter is interpreted as a user name and will probably cause an error.

* The current spool file—one that has been explicitly TEXTed in, or that is current because it is the last spool file processed by the COPY, APPEND, or BROWSE command. If this form is used without a current spool file, an error message is displayed.

username The name of a user on the system. Full MPE wildcarding is supported. This parameter, when used with the optional *acctname*, specifies the set of spool files to be purged. The SPIFF user must have access to files generated by *username*. Refer to "Security".

acctname The name of an account on the system. Full MPE wildcarding is supported. Default: the logon account is assumed. This parameter, when used with the *username*, specifies the set of spool files to be purged. The SPIFF user must have access to files generated by users in *acctname*. Refer to "Security".

Note

Because files in the CREATE state cannot be deleted, SPIFF inserts a STATE <> CREATE in its internal selection equation when processing a *username*[.*acctname*] specification.

seleq A native mode spooler selection equation specifying the set of spool files to delete. The selection equation must be enclosed in brackets as in the following example that purges all spool files created by the user MANAGER.SYS with priority less than 3:

```
PURGE [OWNER=MANAGER.SYS AND PRI<3]
```

If you choose this (*seleq*) form of file set selection, SPIFF inserts an OWNER=!HPUSER.!HPACCOUNT in its internal selection equation, *unless* you explicitly include your own OWNER definition. This prevents

users with SM, OP, or AM capabilities from accidentally accessing files that they did not create.

Because files in the CREATE state cannot be deleted, SPIFF inserts a STATE <> CREATE in its internal selection equation even if you explicitly include your own STATE definition.

Consult one of the following documents for more information about selection equation syntax and semantics:

- *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)
- MPE/iX online help facility

ASK/YES

Once purged, spool files cannot be recovered unless an archive copy has been made. These optional keywords provide an interactive precaution against unintended deletion of spool files. If either keyword is specified with the *spoolfileid* or with *, a warning is displayed and the keyword is ignored.

This parameter is in effect whenever SPIFF is run interactively.

Omitted

The user is presented with one list of all of the spool files selected for purging and is asked to approve the operation. This is an "all or nothing" choice. It is intended for users who are reasonably sure of the set of spool files to be deleted.

Responding Y[ES] purges all of the selected spool files.

Responding with N[O], E[ND], or Ctrl Y exits the command without deleting the spool files.

ASK

The user is presented with each spool file as it is encountered and is asked to approve the deletion.

Responding Y[ES] purges the spool file. Responding N[O] leaves the spool file unchanged.

Responding with E[ND], or Ctrl Y exits the command without deleting the spool file.

YES

SPIFF purges all of the selected spool files without confirmation from the user.

When SPIFF is run non-interactively, it is impossible to conduct a confirmation dialog. Examples of non-interactive operation include running SPIFF in a job or with redirected $STDIN or $STDLIST. In such cases, SPIFF operates as though YES had been specified.

Operation Notes

The PURGE command (abbreviated P) marks one or more spool files for deletion from the system, displaying identifying information (in LISTSPF format) for each marked spool file. The user may specify that SPIFF pause to confirm the deletion(s).

Note that spool files in the CREATE state (OPEN as displayed by SHOWOUT) cannot be purged.

If the spool file is your current text file, it will be closed. If you are the last (or only) accessor, the file is also purged at this time. If the spool file is currently opened by another process, it will not be purged until its last accessor closes it. The resulting display shows the spool file(s) in the DELPND state.

If the PURGE command has paused for a user response to any purge confirmation, entering Ctrl Y terminates the command without purging any more files.

Examples

Assume that spoolid's #O7788 and #O7789 exist, and that the user has access to them:

```
PURGE #O7788

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#O7788     S64      MYFILE     2      1 PP       DELPND       MYUSER.MYACCT

TEXT 7789
PURGE *

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#O7789     S64      MYFILE     2      1 PP       DELPND       MYUSER.MYACCT

PURGE *

You have no current TEXT file. (SPERR 81)
```

The following examples assume that you are MGR.ACCT (the manager of an account), and that you have AM capability.

```
> PURGE USER1.ACCT
  The following spoolfiles have been selected for purging:

SPOOLID       JOBNUM    FILEDES   PRI  COPIES DEV     STATE  RSPFN OWNER

#0452         J265      $STDLIST    2       1  PP     READY        USER1.ACCT
#0781         J518      $STDLIST    2       1  PP     READY        USER1.ACCT
#0779         J514      $STDLIST    2       1  PP     READY        USER1.ACCT
#0782         J520      $STDLIST    2       1  PP     READY        USER1.ACCT
#0784         J524      $STDLIST    2       1  PP     READY        USER1.ACCT

  Purge all spoolfiles shown (Y/[N]) Y

SPOOLID       JOBNUM    FILEDES   PRI  COPIES DEV     STATE  RSPFN OWNER

#0452         J265      $STDLIST    2       1  PP     DELPND        USER1.ACCT
#0781         J518      $STDLIST    2       1  PP     DELPND        USER1.ACCT
#0779         J514      $STDLIST    2       1  PP     DELPND        USER1.ACCT
#0782         J520      $STDLIST    2       1  PP     DELPND        USER1.ACCT
#0784         J524      $STDLIST    2       1  PP     DELPND        USER1.ACCT

> PURGE [(OWNER=USER2.ACCT) AND (PRI <=2)];ASK

SPOOLID       JOBNUM    FILEDES   PRI  COPIES DEV     STATE  RSPFN OWNER

#0557         J354      $STDLIST    2       1  LP     READY        USER2.ACCT
   Purge this spoolfile (Y/[N])? Y

#0569         J365      $STDLIST    2       1  LP     READY        USER2.ACCT
   Purge this spoolfile (Y/[N])? N
   Spoolfile not deleted  (SPWARN 228)

#0612         J394      $STDLIST    2       1  LP     READY        USER2.ACCT
   Purge this spoolfile (Y/[N])? Y

#0613         J394      $STDLIST    2       1  LP     READY        USER2.ACCT
   Purge this spoolfile (Y/[N])? E

>
```

QUIT

Terminates SPIFF and returns control to its parent process.

Syntax

```
> Q[UIT]
```

Operation Notes

The QUIT command (abbreviated Q) terminates the SPIFF process. If a file has been opened by the TEXT command, it is closed before SPIFF terminates. Control is returned to SPIFF's parent process. This is usually a command interpreter, but may be another application if that application supports creation of a child process. In particular, SPIFF may be run from within another copy of SPIFF, using the RUN command. QUITing the second copy of SPIFF returns control to the first instance of SPIFF.

The QUIT command and the EXIT command operate identically.

Note

It is not possible to suspend operation of a child SPIFF process, give control to a parent SPIFF process, and return to the child SPIFF process, as the SPOOK5 program used to do. Each exit of a child process terminates that process and closes all open files associated with that process. Each child process starts afresh when created. As a result, the KILL command is not supported by SPIFF.

SHOW

Displays information about one or more spool files.

Syntax

```
> S[HOW]  [ spoolfileid [ ,spoolfileid [ , ... ] ]
          [ *
          [ [ username[ .acctname ] ] [ ; [ @ ] ] ]
          [ [ seleq               ]   [ [ I ] ] ]
                                      [ [ O ] ]
```

Parameters

none — Displays all spool files (input and output) belonging to the logon user in the logon account.

spoolfileid — An existing spoolid to which the user has access. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (**#**). The full syntax is [#O]*nnnnn* or [#I]*nnnnn*, where the *n*'s represent digits. If the **#** is used, the `O` or "I" must also be used. If the `O` or "I" is used without the **#**, the parameter is interpreted as a *username* and will probably cause an error. Default: All output spool files created by the logon user in the logon account.

* — The current spool file—one that has been `TEXT`ed in, or is current because it is the last spool file processed by the `COPY`, `APPEND`, or `BROWSE` command. If this form is used when no spool file has been `TEXT`ed, an error message is displayed.

When this parameter is specified, the current spool file is displayed using `LISTSPF`'s `DETAIL` (two-line) display format.

username — The name of a user on the system. Full MPE wildcarding is supported. The `SPIFF` user must have access to files generated by *username*. Refer to "Security".

acctname — The name of an account on the system. Full MPE wildcarding is supported. Default: the logon account is assumed. The `SPIFF` user must have access to files generated by users in *acctname*. Refer to "Security".

seleq — A native mode spooler selection equation specifying the set of spool files to be shown. The selection equation must be enclosed in brackets as in the following example that displays all spool files created by the user `MANAGER.SYS` with priority less than 3:

```
SHOW [OWNER=MANAGER.SYS AND PRI<3]
```

If you choose this (*seleq*) form of file set selection, SPIFF inserts an OWNER=!HPUSER.!HPACCOUNT in its internal selection equation, *unless* you explicitly include your own OWNER definition. This prevents users with SM, OP, or AM capabilities from accidentally accessing files that they did not create.

Consult one of the following documents for more information about selection equation syntax and semantics:

- *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)
- MPE/iX online help facility

`@` Displays selected files using LISTSPF's DETAIL (two-line) format. Omitting this parameter displays selected files in LISTSPF's one-line format. @ is valid only when selecting files with *username.acctname* or with *seleq*.

`I` Restricts display to input spool files. I is valid only when selecting files with *username.acctname* or with *seleq*.

`O` Restricts display to output spool files. O is valid only when selecting files with *username.acctname* or with *seleq*.

Note

You may specify both I and O to display both input and output spoofiles. This is equivalent to specifying neither.

Operation Notes

The SHOW command (abbreviated S) displays attributes of one or more spool files using the MPE/iX LISTSPF command format. The display may be in either the one-line or two-line format of the LISTSPF command, according to your specification of the SHOW command:

1. The two-line format is produced in either of these cases:

   SHOW *spoolfileid* (or a list of *spoolfileids*), or SHOW *.

   SHOW *username.acctname* or *seleq* with the ;@ option.

2. The one-line format is produced in all other cases.

When the *username.acctname* or *seleq* form is used, the @, I, and O keywords may be used in any combination or repeatedly. Once used, each parameter enables its option for the duration of the command and cannot be disabled. Thus @IOI@OO@I and @@@@@@@@ are both valid.

Examples

```
SHOW ;0

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER

#07817     J166     $STDLIST   2      1 LP       CREATE       MYUSER.MYACCT
#07820     S104     MYFILE     2      1 PP       READY        MYUSER.MYACCT

INPUT SPOOL FILES           OUTPUT SPOOL FILES
ACTIVE   = 0;               CREATE   = 1;              READY    = 1;
OPEN     = 0;               DEFER    = 0;              SELECTED = 0;
READY    = 0;               DELPND   = 0;              SPSAVE   = 0;
                            PRINT    = 0;              XFER     = 0;
                            PROBLM   = 0;

TOTAL IN FILES   = 0;       TOTAL OUT FILES   = 2;
      IN SECTORS = 0;             OUT SECTORS = 2128;

OUTFENCE = 6

> SHOW *

You have no current TEXT file. (SPERR 81)
> TEXT 7820
> SHOW *

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER
           FORMID   JOBNAME     COPSRM SECTS    RECS   PAGES DATE     TIME

#07820     S104     MYFILE     2      1 PP       READY        MYUSER.MYACCT
                                      1 80       234       ~4 01/07/92 08:42

INPUT SPOOL FILES           OUTPUT SPOOL FILES
ACTIVE   = 0;               CREATE   = 0;              READY    = 1;
OPEN     = 0;               DEFER    = 0;              SELECTED = 0;
READY    = 0;               DELPND   = 0;              SPSAVE   = 0;
                            PRINT    = 0;              XFER     = 0;
                            PROBLM   = 0;

TOTAL IN FILES   = 0;       TOTAL OUT FILES   = 1;
      IN SECTORS = 0;             OUT SECTORS = 80;

OUTFENCE = 6
```

```
> SHOW MYUSER.MYACCT;@0

SPOOLID    JOBNUM   FILEDES  PRI COPIES DEV      STATE  RSPFN OWNER
           FORMID   JOBNAME      COPSRM SECTS    RECS   PAGES DATE     TIME

#07817     J166     $STDLIST   2      1 LP       CREATE       MYUSER.MYACCT
                    MYJOB             1 2048     0

#07820     S104     MYFILE     2      1 PP       READY        MYUSER.MYACCT
                                      1 80       234       ~4 01/07/92 08:42

INPUT SPOOL FILES            OUTPUT SPOOL FILES
ACTIVE   = 0;                CREATE   = 1;              READY    = 1;
OPEN     = 0;                DEFER    = 0;              SELECTED = 0;
READY    = 0;                DELPND   = 0;              SPSAVE   = 0;
                             PRINT    = 0;              XFER     = 0;
                             PROBLM   = 0;

TOTAL IN FILES   = 0;        TOTAL OUT FILES   = 2;
      IN SECTORS = 0;              OUT SECTORS = 2128;

OUTFENCE = 6
>
```

STORE

Stores one or more files to tape using the MPE/iX `STORE` subsystem.

Syntax

```
> ST[ORE] { spoolfileid [ ,spoolfileid [ , ... ] ] }
          { username[ .acctname ]                   }
          { seleq                                   }

[ ;store_option [ ;store_option [ ; ... ] ] ]
```

Parameters

spoolfileid An existing output spoolid to which the user has access. To be taken as a spoolid (instead of a *username*), this parameter must begin with a number or with a pound sign (`#`). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the `#` is used, the `O` must also be used. If the `O` is used without the `#`, the parameter is interpreted as a *username* and will probably cause an error.

username The name of a user on the system. Full MPE wildcarding is supported. The `SPIFF` user must have access to files generated by *username*. Refer to "Security".

acctname The name of an account on the system. Full MPE wildcarding is supported. Default: the logon account is assumed. The `SPIFF` user must have access to files generated by users in *acctname*. Refer to "Security".

seleq A native mode spooler selection equation specifying the set of spool files to store. The selection equation must be enclosed in brackets as in the following example that stores all spool files created by the user `MANAGER.SYS` with priority less than 3:

```
STORE [OWNER=MANAGER.SYS AND PRI<3]
```

If you choose this (*seleq*) form of file set selection, `SPIFF` inserts an `OWNER=!HPUSER.!HPACCOUNT` in its internal selection equation, *unless* you explicitly include your own `OWNER` definition. This prevents users with SM, OP, or AM capabilities from accidentally accessing files that they did not create.

Consult one of the following documents for more information about selection equation syntax and semantics.

- *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115)
- MPE/iX online help facility

store_option Any MPE/iX `STORE` subsystem option. The first option should specify either an output device, such

as a tape drive, or be left empty (but with the semicolon to mark its position). Refer to the *Storing Files and Backing Up the System Reference Manual* (32650-90140) for a full list of available options.

Operation Notes

SPIFF's STORE command (abbreviated ST) is an interface to the MPE/iX STORE subsystem that allows fileset selection by specific *spoolfileid*(s), *username.acctname*, or *seleq*. This augments normal MPE/iX STORE fileset wildcarding with attributes specific to spool files.

The chosen set of spool files are listed to an internal temporary indirect file, which is then passed to the MPE/iX STORE subsystem along with any *store_options*, such as ;SHOW.

Examples

```
STORE MYUSER.MYACCT;*MYTAPE;SHOW
```

(Output from STORE's SHOW option is not listed here.)

```
STORE [DATE < 01/08/92 AND OWNER=MYUSER.MYACCT];;SHOW;PURGE
```

Notice the empty tape file specifier. (Output from STORE's SHOW option is not listed here.)

TEXT

Accesses an output spool file for use by the ALTER, APPEND, BROWSE, COPY, FIND, LIST, PURGE, and SHOW commands.

```
> T[EXT] { spoolfileid }
         { *          }
```

Parameters

spoolfileid — An existing spoolid to which the user has access. This parameter must begin with a number or with a pound sign (#). The full syntax is [#O]*nnnnn*, where the *n*'s represent digits. If the # is used, the O must also be used. If the O is used without the #, an error message is displayed.

* — The current spool file—one that has been TEXTed in, or is current because it is the last spool file processed by the COPY, APPEND or BROWSE command. If this form is used when no spool file has been TEXTed, an error message is displayed.

Operation Notes

The TEXT command (abbreviated T) opens the specified output *spoolfileid* for use as the currently TEXTed spool file. If another spool file has been TEXTed in previously, it is first closed.

TEXTing in a spool file enables the * parameter of the ALTER, APPEND, BROWSE, COPY, PURGE, SHOW, and TEXT commands. Once * is enabled, you may, if you wish, omit a *spoolfileid* specification in the APPEND, BROWSE, and COPY commands. A currently TEXTed spool file is required for the FIND and LIST commands.

The TEXT * form of the command closes the current spool file without TEXTing in another. It is an error to use this form if there is no current spool file.

Other issues to be aware of include these:

- Native mode spool files have no LOCKED state; therefore, there is no indication that a spool file has been opened with the TEXT command. The spool file state does not change as a result of its being opened by SPIFF.
- The PURGE * command close the current spool file as part of its operation, allowing the PURGE to complete. This is consistent with SPOOK5 behavior.
- You may TEXT in an output spool file in the CREATE state. Once it is the current spool file, you may execute the APPEND, COPY, FIND, and LIST commands on it. Each such command obtains the current end-of-file before evaluating its *range* expression. The BROWSE, INPUT, OUTPUT, and STORE commands are not allowed; they require exclusive access to the file. The PURGE command is not allowed on a file that is in the CREATE state.

- The last spool file processed by the APPEND, BROWSE, or COPY command is left open as the current spool file.

XPLAIN

Displays a summary of SPIFF commands.

Syntax

```
> X[PLAIN]
```

Parameters

The X[PLAIN] command has no parameters.

Operation Notes

The XPLAIN command (abbreviated X) is implemented as the HELP SUMMARY command. This serves two purposes:

- It retains SPOOK5's XPLAIN command in the form most nearly compatible with that command.
- It provides a simple interface to that form of help for users who would rather not concern themselves with some of the subtleties of the revised HELP command.

Refer to the HELP command description for further details.

Spool File Transfer Utility (SPFXFER)

The spool file tape transfer utility (SPFXFER) transports spool files between MPE/iX systems containing the native mode spooler and MPE systems that do not. The SPFXFER utility is located in the PUB group of the SYS account. This utility can do the following:

- transfer NMS spool files to tape in a format that SPOOK and SPOOK5 (on MPE V/E releases G.02.B0 and later) can read
- read SPOOK tapes from any release of MPE/iX and MPE V/E

You can use SPFXFER to transport spool files between any two NMS systems, also. Generally, though, use STORE and RESTORE. Information about STORE and RESTORE is in chapter 2 of this manual.

SPFXFER has four commands. They are: HELP, INPUT, OUTPUT, and EXIT. You may truncate each command at any point. For example, E, EX, EXI, and EXIT are all valid forms of the EXIT command.

To run SPFXFER, enter:

```
RUN SPFXFER.PUB.SYS
```

or if SPFXFER is in the search path defined by your HPPATH system variable, enter:

```
SPFXFER
```

Most MPE/iX commands and user defined commands (UDCs) may be executed at the SPFXFER prompt (>) by preceding the command with a colon (:). For example, to execute the MPE/iX command LISTSPF within SPFXFER, enter:

```
SPFXFER
:LISTSPF
```

Listing the commands

The HELP command lists all SPFXFER commands and their syntax. The syntax of the command is as follows:

```
HELP
```

It has no parameters. To use the command, enter:

```
SPFXFER
 HELP
```

Restoring spool files onto the system disk

The INPUT command reads output spool files from tape onto a system disk. The user and account of the owner need not exist in the system directory, nor does the INPUT command create them. The system places the spool files in the OUT.HPSPOOL account and assigns them new SPOOLIDs (spool file identification numbers). The spool files are linked to the spooling subsystem.

Note The system does not restore user labels because NMS spool files do not have them.

For each restored spool file, `SPFXFER` displays:

- the old and the new spool file identifications
- new job or session number in the format #J*nnnn* or #S*nnnn* (The single quote distinguishes these spool files from those originally created on this system.)
- the target device or class name
- the owner

A restored spool file has the same output priority as it did before it was stored on tape.

The syntax for `INPUT` is:

`INPUT` [[*username*[. *accountname*]] ; *tapefile*
[*spoolid*[, *spoolid*] ...]

Parameter definitons

username	The user name of the creator of the files or `@` to designate all users.
accountname	The account name of the creator of the files or `@` to designate all accounts.
spoolid	The spool file device identification (in the format [#O]*nnnn* where *nnnn* is a number) given by the system that stored them. On CM and MPE V/E systems, *spoolid* has a maximum value of 32,767.

Note If you omit both *username.accountname* and *spoolid*, the system restores all files belonging to the logon user.

tapefile	The name of the file that denotes the tape device that contains the spool files. A file equation must exist for the tape device. You must backreference *tapefile* with an asterisk (*) in the form **tapefile*.

To restore spool files #O100 and #O200 from a tape formally designated in a file equation as T, mount the tape, put it online, and enter:

```
FILE T;DEV=TAPE
SPFXFER
 INPUT #O100,#O200 ;*T
```

or

```
FILE T;DEV=TAPE
SPFXFER
 INPUT 100,200 ;*T
```

A message like the following example appears on the console:

```
?10:41/#559/45/LDEV# FOR "T" ON TAPE (NUM)?
```

If you are the console user, respond with CTRL A. An "=" sign appears. Enter:

```
REPLY 45,7
```

where 45 is the process identification number (PIN) and 7 is the LDEV for the tape drive for this example.

Note

NMS spool files can be up to 4 Gbytes long. Even a spool file smaller than this may not fit on MPE systems not containing the NMS if the spool file exceeds 32 extents of the size configured on the target system.

Storing linked spool files onto tape

The OUTPUT command stores linked output spool files from OUT.HPSPOOL to tape in a SPOOK-compatible format. Then you can restore the spool files to an MPE/iX system containing NMS using the SPFXFER INPUT command or to a MPE/iX system not containing NMS or an MPE V/E system using the SPOOK INPUT command.

Note

Because NMS spool files do not have user labels the SPFXFER utility does not store spool file user labels on tape.

For each stored spool file, SPFXFER displays the current spool file identification, the job or session number of the file's owner, the logical device class or name, the number of sectors in the file, and the owner's name and account.

If the job or session number that MPE/iX assigns is larger than 16,383, the system assigns a number not to exceed 16,383. Also, the system reduces the number of copies to 127 if the total number of copies exceeds this. This change is due to backward compatibility requirements for tapes that the SPOOK utilities of MPE/iX systems not containing NMS and of MPE V/E systems create.

The syntax for OUTPUT is:

```
OUTPUT [ [ username [ .accountname ] ]  ]
       [ [ spoolid [ ,spoolid ] ... ]   ] ;*tapefile [ ;PURGE ]
```

Parameter definitions

username	The user name of the creator of the files or @ to designate all users.
accountname	The account name of the creator of the files or @ to designate all accounts.
spoolid	The spool file identification number in the form [#O]*nnnn* of the spool files. It has a maximum value of 9,999,999. If the *spoolid* of any spool file being stored by using OUTPUT exceeds 32,767, the system assigns a SPOOLID less than 32,767.

Note

If *username.accountname* and *spoolid* are both omitted, the system stores all files belonging to the logon user.

tapefile	The name of the file that denotes the tape device which stores the spool files. A file equation must exist for the tape device. You must backreference *tapefile* with an asterisk in the form **tapefile*.
PURGE	This option tells the system to purge the spool files after they are stored on tape.

To store all linked spool files created by USER whose account is ACCOUNT on a magnetic tape formally designated as MYTAPE in a file equation and then to delete the spool files, enter:

```
FILE MYTAPE;DEV=TAPE
SPFXFER
 OUTPUT USER.ACCOUNT ;*MYTAPE ;PURGE
```

Like the example for the INPUT command, the tape is mounted, the drive is online, and a reply must be given before the files are stored to tape. In addition, the tape must have a write ring.

Additional information about the INPUT and OUTPUT commands.

- To use the INPUT and OUTPUT commands, you must have nonshareable devices (ND) capability to access the tape drive and system manager (SM) or system supervisor (OP) capability; otherwise, you can only use the HELP and EXIT commands.
- Before you use the INPUT and OUTPUT commands, set up a file equation as follows:

  FILE *formaldesig* [=*filereference*] ;DEV = *device*

  where *formaldesig* is the formal file designator and *device* is the device class name or the logical device number of a magnetic tape unit. For example, to restore all spool files from a tape named

TAPEFILE mounted on a tape drive whose device class name is TAPE, enter:

```
FILE TAPEFILE; DEV=TAPE
SPFXFER
 INPUT @ ;*TAPEFILE
```

- If the system restores files from or stores files to multiple tape reels, the operator must respond to the following prompt:

  CHANGE REELS ON LDEV *nnn*? (Y / N)

  Entering N (for "no") aborts the operation. Entering any other key means "yes," and the operator must mount another tape.

  For the INPUT command, if the operator mounts a wrong reel, the following prompt appears:

  INCORRECT REEL - TRY AGAIN? (Y / N)

  The prompt is displayed until either the operator mounts a correct reel or enters N. If the response is N, the system aborts the INPUT operation.

- If I/O errors are encountered while reading from or writing to disk or tape, then SPFXFER displays the spool file identification, the SPFXFER error number, and the file system error number.

The SPFXVAR Variable

The most recent SPFXFER error number is placed in the CI variable SPFXVAR. For example:

```
GATO
Invalid command syntax.
Native Mode Spooler Message 10755
:SHOWVAR SPFXVAR
SPFXVAR=-10755
```

Leaving SPFXFER

The EXIT command terminates SPFXFER.

The syntax for the command is as follows:

```
EXIT
```

It has no parameters. To use the command, enter:

```
EXIT
```

Chapter 2 of this manual contains more examples of SPFXFER.

The Print Spool File Utility (PRINTSPF)

The print spool file utility (PRINTSPF) displays both the data and the special overhead area of each record of a spool file. The syntax of PRINTSPF is similar to that of the MPE/iX PRINT command. You can use PRINTSPF to print ranges of records (both absolute and relative to the EOF) and to display the record number of each record.

Although the primary purpose of PRINTSPF is to display formatted spool file records, you may use it to display other file types as well. If you use PRINTSPF to display a file that is not a spool file, it displays in the same manner as the PRINT command.

To stop the display, use CTRL Y.

Note

Using CTRL Y to stop the display also terminates PRINTSPF and returns you to the CI prompt. You must start PRINTSPF again in order to resume the display.

Only users with SM capability can display private spool files.

Note

The PRINTSPF utility is in PUB.SYS. Make sure that PUB.SYS is in your search path before using PRINTSPF. To add PUB.SYS to your search path, enter:

```
SETVAR HPPATH HPPATH+",PUB.SYS"
```

To view your search path, enter:

```
SHOWVAR HPPATH
```

The syntax for PRINTSPF is shown below:

PRINTSPF *filename*

or

PRINTSPF "[FILE=]*filename*

[[;START=]*startrec*]

[[;END=]*endrec*]

[[;WIDTH=]*linewidth*]

[;NUM]"

The second form of the PRINTSPF syntax follows the more flexible MPE/iX command line syntax. Refer to the *MPE/iX Commands Reference Manual* (32650-90003). When more than one token is specified, the double quotes are required.

Parameter definitions

filename — Actual name of the file to be printed to $STDLIST. The system ignores file equations unless an asterisk (*) precedes the file name, indicating a backreference. If the file name is of the form I*nnnn* or O*nnnn*, then PRINTSPF searches the current logon directory for the spool file first. If it does not find the file there, PRINTSPF then searches IN.HPSPOOL for I*nnnn* or OUT.HPSPOOL for O*nnnn*. This searching procedure is done only if you do not qualify the file name with a group or account name.

startrec — Specifies the record number of the first file record to be displayed. A positive *startrec* is relative to one. A negative *startrec* is relative to the end-of-file. For example, *startrec* of -5 indicates the fifth record from the end of the file. Zero is changed to one. The default is the first record of the file.

endrec — Specifies the last record of the file to be displayed. A positive *endrec* is relative to one. A negative *endrec* is relative to the end-of-file. For example, *endrec* of -5 indicates the fifth record from the end of file. Zero is changed to one. The default is the last record in the file.

linewidth — Specifies the number of characters of the formatted record to display on each line. The default is the record size of $STDLIST minus one. If the line contains more characters than *linewidth*, the line is truncated.

NUM — Specifies numbering the displayed lines. The numbers appear before the displayed lines. The number displayed is the actual record number, relative to one, of each displayed line. The default is not to display record numbers.

Output format

There are five types of spool file records: open, write, close, control, and device control. For each of these record types, PRINTSPF displays a line of formatted data. The following are the formats for each type of record.

OPEN

```
OP P1=$xxxx P2=$xxxx [BUF/#llll= aaa...a]
```

where *xxxx* are hex values for P1 and P2, *llll* is the buffer length in decimal, and *aaa...a* is the ASCII contents (forms message) of the buffer.

CLOSE

```
CL P1=$xxxx P2=$xxxx
```

where *xxxx* are the hex values for P1 and P2.

WRITE

```
WR P1=$xxxx P2=$xxxx [CC=%cc] [BUF/#llll=aaa...a]
```

where *xxxx* are the hex values for P1 and P2, *cc* is the octal value of the carriage control, *llll* is the buffer length in decimal, and *aaa...a* is the ASCII contents (data) of the buffer.

CONTROL

```
CO P1=$xxxx P2=$xxxx
```

where *xxxx* are hex values for P1 and P2.

DEVICE CONTROL

```
DC P1=$xxxx P2=$xxxx FN=#ffff [BUF/#llll=hhhh]
```

where *xxxx* are the hex values for P1 and P2, *ffff* is the decimal value of the FDEVICECONTROL function, *llll* is the buffer length in decimal, and *hhh...h* is the hex contents of the buffer.

Suppose you are the creator of spool file O2.OUT.HPSPOOL. You want to display 77 characters per line and the first ten records. Enter:

```
PRINTSPF "O2; END=10; WIDTH=77"
```

The output appears in figure 6-1.

```
PRINTSPF A.00.00 Copyright (C) Hewlett-Packard 1989 All rights reserved.
OP P1=$0000 P2=$0000
WR P1=$0001 P2=$0000 CC=%000 BUF/#  21= :JOB MANAGER.SYS,PUB.
WR P1=$0001 P2=$0000 CC=%000 BUF/#  51=PRIORITY = DS; INPRI = 8; TIME = UNLI
WR P1=$0001 P2=$0000 CC=%000 BUF/#  17= JOB NUMBER = #J1.
WR P1=$0001 P2=$0000 CC=%000 BUF/#  28= MON, NOV 22, 1989,  9:35 AM.
WR P1=$0001 P2=$0000 CC=%000 BUF/#  25= HP3000 / MPE XL Z.34.45
WR P1=$0001 P2=$0000 CC=%000 BUF/#  75= MPE XL CI A.20.00 Copyright (C) Hewle
WR P1=$0001 P2=$0000 CC=%000 BUF/#
WR P1=$0001 P2=$0000 CC=%000 BUF/#  23= :setvar hpautocont true
WR P1=$0001 P2=$0000 CC=%000 BUF/#   1= :
```

Figure 5-1. PRINTSPF **Sample Output**

To display the entire contents of spool file O2.OUT.HPSPOOL, enter:

```
PRINTSPF O2
```

Note

If file O2 exists in your logon group and account, PRINTSPF prints that file instead of O2.OUT.HPSPOOL.

Chapter 2 contains other examples for PRINTSPF.

Error messages

The following are error messages of `PRINTSPF`:

- Command syntax error.
- The file name is too long.
- No file name was specified.

A

Spool File Block Format (SBF)

All regular MPE disk files have one of three logical record formats: fixed length, variable-length, or undefined length. We usually speak of a file in terms of its record format, for example, "variable-length file". Spool files are variable-length files with additional overhead. Here is a quick review of the essential elements of any MPE variable-length file.

When you specify a record length and/or blocking factor for a variable length file, you tell the file system how large the physical record should be. For MPE V/E systems, the physical record determines how disk storage is allocated. We limit the present discussion to disk files. For MPE/iX systems, disk storage is allocated differently, so the physical record is now more of an abstraction that defines a fixed limit to variable-length records. In practice this means that if the next logical record to be added to the file does not fit in the current physical record, a new physical record must be started for it.

Table A-1 shows a typical physical record consisting of several variable-length logical records.

Table A-1. Physical Record (Block)

byte count |. . . . .
. . . . . | byte count |. . . . .
⋮
. . . . | byte count |. . . . .
. . . . . | -1 |////////

Because each record may have a different length, when you write a record to the file, the file system prefixes the record with a 16-bit byte count. The byte count represents the number of data bytes in the record, excluding itself and (for ASCII files) any unused odd byte at the end of the record. Within a given physical record, the file system uses these byte counts to find the start of the next logical record relative to the start of the current one.

Logical records never span a physical record boundary, so eventually there comes a time when you write a record that does not fit in the space remaining in the current physical record. When the file system discovers this situation, it writes a special byte count of -1 (octal 177777), then writes your record at the start of a new physical record. The space between the -1 and the end of the physical record is not used. (Later, when you read the file, the -1 alerts the file system that there is no more data in the current physical record and that it should retrieve your next record from the next physical record).

A spool file is a special case subset of a variable-length file:

- Its physical record (block) length is always 1K (1024) bytes.
- It is always an ASCII (never a binary) file.
- In addition to the byte count shown above, each record contains four more 16-bit fields (eight more bytes) of overhead (described below).
- The first byte of the data portion of the record may be a carriage control (CCTL) byte; if so, it too is considered overhead.
- The last four bytes of each physical record contain a record count, thereby shortening by four the number of bytes available for spool file records.

Table A-2 displays a typical spool file block.

Table A-2. Spool File Block

physical byte count | logical byte count | func | P1 | P2 |.	
. . . . . .| physical byte count | logical byte count | func | P1 | P2 |.	
⋮	
. . . . | physical byte count | logical byte count | func | P1 | P2 |.	
. . . . . | -1| ////////////////// | 1st record of block	

Field	Description
Physical byte count	Same as the "byte count" shown in table A-1. The new name distinguishes it from "logical byte count".
Logical byte count	The original byte count of the data portion of the spool file record (nominally everything between the P2 field and the physical byte count field of the next record). It was included originally so that trailing blanks of a record could be left out of the spool file and reinserted at some lower printing level such as the printer driver, if necessary. In practice it has not been used this way, so its value is usually eight less than the physical byte count (remember that the physical byte count does not include itself). Further discussion is beyond the scope of this description.
Func, P1, P2	These are low-level printer parameters. Originally they were part of the MPE V/E privileged interface to `ATTACHIO`. Although the native mode spooler does not use `ATTACHIO`, the position and function of these parameters have been retained. A complete description is beyond the scope of this appendix, but generally speaking, FUNC is a printer function code (such as write or control), while P1 and P2 ("P" for "parameter") vary with function code and further define the operation to be performed.
First record of block	This is an ordinal whose value is the record number (relative to the start of user data in the spool file; 0 is the first record) of the first record of the current spool file block. For example, if the first block of user data consists of 8 records (0 through 7), the field at the end of the first block is 0 and the field at the end of the second block is 8 (because record 8 is the first record in the second block). This field was added to MPE V/E spool files to allow pseudo-direct access to any record in the spool file. It has been retained in native mode spool files. Ordinarily, variable-length files must be read serially. This means that to display record 10000 you have to read and discard records 0-9999; however, using multirecord NOBUF access and binary search techniques, this field allows you to access record 10000 much more rapidly.

PARAMETER	VALUE	MEANING
FUNC	1	Write
P1	1	The first byte of the data portion of the record is a carriage control (CCTL) byte, usually the one specified by a user in his/her FWRITE intrinsic call. It is not part of the user data and should not be displayed on the printer.
P1	<> 1	P1 is itself the CCTL byte. The first byte of the data portion is part of the user data and should be displayed.
P2	Bitmask	If the least significant bit (15) is 1, the record is to be printed in prespace mode. If it is 0, the record should be printed in postspace mode. See the note below. If bit 14 is 1, certain printer operations are carried out without regard for the perforation area separating two sheets of Z-fold paper. If bit 14 is 0, these operations skip over the perforation area, if necessary. Refer to the *MPE/iX Intrinsics Reference Manual* (32650-90028) for details. All other bits are reserved.

Figure A-1. Example: FUNC, P1, P2

In postspace mode, the data in the record is sent to the printer before the CCTL byte is interpreted and sent. In prespace mode, the CCTL byte is interpreted and sent first, followed by the data.

B

Spooler Command Comparison

The MPE/iX (version B.40.00 and later versions) native mode spooler (NMS) is a complete native mode replacement of the previous MPE XL spooling subsystem. The utility SPOOK no longer exists. Instead, SPOOK commands are replaced with command interpreter (CI) commands, the PRINTSPF and SPIFF utilities, the SPFXFER utility, STORE, RESTORE, and editor subsystems.

The NMS commands include the following:

```
LISTSPF
SPOOLER
SPOOLF
```

The SPOOLER command may be entered only at the master console unless allowed to other users through the ALLOW and ASSOCIATE commands. The LISTSPF and SPOOLF commands work according to your capabilities and whether you are the console user or not. Chapter 2 has tasks and examples which demonstrate this.

The MPE/iX commands that can be used with spool files are the following:

```
BUILD          OPENQ
COPY           OUTFENCE
FILE           PURGE
FORMSALIGN     RENAME
JOB            SHUTQ
LISTEQ
```

The following MPE/iX commands available for spool file/spooler control, prior to version A.40.00, are still supported and function almost exactly the same as they always have:

```
STARTSPOOL           OPENQ
STOPSPOOL            SHUTQ
SUSPENDSPOOL         OUTFENCE
RESUMESPOOL          HEADON
ALTSPOOLFILE         HEADOFF
DELETESPOOLFILE
SHOWIN
SHOWOUT
```

This appendix provides a series of tasks and the appropriate MPE/iX commands. The examples illustrate the NMS commands and other CI commands. The complete syntax for the NMS commands is in chapter 4.

NMS spool files are variable-length files that are identified by a special file type. Input and output spool files are automatically created in the groups `IN.HPSPOOL` and `OUT.HPSPOOL`, respectively. An input spool file is linked to the spooling subsystem until the job has logged off, been aborted, or the `DATA` file has been read by a user process.

An output spool file is linked to the spooling subsystem until the last copy has been printed. Then the spool file is deleted unless it is saved with the command `SPOOLF ... ;ALTER ;SPSAVE`. It is also possible to save a spool file by using a file equation (the `FILE` command with the `;SPSAVE` option) or by using the `;SPSAVE` option with the `JOB` command.

Two spool file directories, referred to collectively as the `SPFDIR`, contain all information about input and output spool files. A spool file is *linked* to the spooling subsystem if the spool file has an entry in the SPFDER.

Because spool files are ordinary disk files, they are not lost during system boots. File recovery is no more complicated for spool files than for other permanent disk files.

A new checkpoint file is a companion to an output spool file. The checkpoint file helps the spooler recover from device problems like power failure and paper jams. When a spool file does not print completely, the next spooler process that prints it on the same device uses the checkpoint file.

You may designate an output spool file as a private file, a file that other users can not access. You cannot save or copy private files, but you may purge, print, or (within limits) alter them by using the `SPOOLF` command.

Altering a Spool File

To alter the attributes of a spool file, you may use either of these approaches:

- Use the command `ALTSPOOLFILE`, which allows you to change the number of copies to <= 127.
- Use command `SPOOLF` with the parameter `ALTER`, which allows you to change the number of copies to > 127.

The command `ALTSPOOLFILE ... ;DEFER` changes a file's priority to 0 and leaves the file in the Ready state. The `SPOOLF` command with the `;DEFER` parameter places the file in the deferred state but leaves the priority unchanged.

To change the priority of output spool file O2050 to 9, enter:

```
SPOOLF O2050;ALTER;PRI=9
```

or

```
SPOOLF O2050;PRI=9
```

To save spool file O2050 in the HPSPOOL account after the last copy has printed, enter:

```
SPOOLF O2050;ALTER;SPSAVE
```

or

```
SPOOLF O2050;SPSAVE
```

To change to a total number of copies greater than 127, use SPOOLF. For example, to change the number of copies of spool file O2050 to 130, enter:

```
SPOOLF O2050;ALTER;COPIES=130
```

or

```
SPOOLF O2050;COPIES=130
```

Deleting a Spoolfile

To delete a spool file, use one of these:

- the command SPOOLF with the parameter ;DELETE
- the command PURGE or the command DELETESPOOLFILE

If it can gain access to the file, the PURGE command purges nonprivate output spool files. The DELETESPOOLFILE and SPOOLF ... ;DELETE commands purge private output files and nonprivate output files. (In general, you cannot copy, browse, open, store, or alter private files. They are a security mechanism and are discussed in chapter 2.) PURGE and DELETESPOOLFILE have not changed.

You may delete private *input* spool files using SPOOLF ... ;DELETE or DELETESPOOLFILE only if the input spool files are DATA files in the Ready state. Job input spool files can only be deleted with the ABORTJOB command.

To delete output spool file O2050, enter:

```
SPOOLF O2050; DELETE
```

Stopping and Resuming Header and Trailer Output

Use the commands HEADOFF and HEADON. They remain the same.

Header and trailer information pages are the data pages that appear before and after a printed file, but that are not part of the file's text.

To stop header/trailer output to logical device 6, enter:

```
HEADOFF 6
```

To resume header/trailer output to logical device 6, enter:

```
HEADON 6
```

Both HEADON and HEADOFF take effect between copies of a spool file.

Enabling and Disabling Spooling

Use the NMS command SPOOLER with the parameter OPENQ and the NMS command SPOOLER with the parameter SHUTQ or the commands OPENQ and SHUTQ. [DEV=] *devname* and ;SHOW are parameters for OPENQ and SHUTQ. *Devname* is the device name of the spoolable device and is configured with the SYSGEN utility.

These commands enable and disable spooling and spool queues for a specified device, device name, or all devices belonging to a device class.

To shut the spool queue for device name LDEV6, enter:

```
SHUTQ LDEV6
```

or

```
SPOOLER LDEV6;SHUTQ
```

To open the queue for logical device 6 and to show the state of the queue and other information about logical device 6, enter:

```
OPENQ 6;SHOW
```

or

```
SPOOLER 6;OPENQ;SHOW
```

SHUTQ @ disables all spooling queues that are currently open and inhibits the creation of any output spool files.

OPENQ @ opens *all spooling queues* that were open *at the time of the last* SHUTQ @ or that have been opened since then and have not been shut. It *does not* open all queues on the system. You can think of OPENQ @ as a global enabling of all device queues that are already open.

Suppose OPENQ @ is in effect (the default at boot time). Consider the following sequence:

```
OPENQ 6    (Spools data to LDEV 6.)
OPENQ 11   (Spools data to LDEV 11.)
SHUTQ @    (Cannot spool data to any device)
OPENQ 19   (Still cannot spool data to any device)
SHUTQ 11   (Still cannot spool data to any device)
OPENQ @    (Spools data to LDEVs 6 and 19 but not
                LDEV 11 since LDEV 11's spool queue was
                shut with the previous command)
```

Controlling the Processing of Output Spool Files

Use OUTFENCE. Its ;DEV= parameter allows you to specify logical devices, device classes, or device names. Omit the ;DEV= parameter to set a systemwide outfence. When you set a systemwide outfence, all device-specific outfences are cleared. Following any kind of system startup, the default system-wide outfence is 14.

This command defines the minimum priority an output spool file must have in order to be printed. A spool file must have a priority greater than the outfence in order to print.

To set the outfence to 8 for all logical devices in device class EPOC, enter:

```
OUTFENCE 8;DEV=EPOC
```

To set the outfence to 5 for device name LDEV6, enter:

```
OUTFENCE 5;DEV=LDEV6
```

The default system-wide outfence is 14 for each type of system startup.

Starting Spooling

Use the NMS command SPOOLER with the parameter START or the command STARTSPOOL. These commands create and activate a new spooler process. By default, these commands also open the spooling queue(s) for the device(s).

Both commands initiate spooling for *all* devices in a device class if you specify the device class.

To initiate spooling for device class LP, which contains three devices 6, 11, and 19, enter:

```
SPOOLER LP;START
```

A spooling process is created for each of the devices 6, 11, and 19 and, by default, spooling queues are opened for each of the three devices.

Stopping Spooling

Use the NMS command SPOOLER with the parameter STOP or the command STOPSPOOL. By default, these commands also close the spooling queues for the device(s).

Both commands stop spooling for *all* devices in a specified device class. To stop spooling for device class LP, which contains three devices 6, 11, and 19, enter:

```
SPOOLER LP;STOP
```

The spooling process is stopped for each of the devices 6, 11, and 19. By default, the spooling queues for the three devices are closed.

Suspending Spooling

Use the command SPOOLER with the parameter SUSPEND or the command SUSPENDSPOOL.

SUSPENDSPOOL and SPOOLER ... ;SUSPEND without the ;FINISH option causes the spooler process to retain ownership of the spool file it is currently processing. Furthermore, the spooler does not cap the current file with a page eject and trailer, but pauses the output. The printer resumes printing exactly where it left off when the SPOOLER ... ;RESUME command with no offset is issued.

Both SPOOLER ... ;SUSPEND ;KEEP and SPOOLER ... ;SUSPEND with no offset pause the output. Again, the spooler resumes spooling exactly where it left off.

Caution

When using network printers, avoid using the following commands to suspend the spooler in mid-file:

```
SPOOLER ... ; STOP
SPOOLER ... ; SUSPEND; OFFSET=anything
SPOOLER ... ; RESUME; OFFSET=anything
SPOOLER ... ; SUSPEND; NOKEEP
SPOOLER ... ; RELEASE
```

Many interfaces drop a network connection if the printer is ready to receive data but no data is being sent within a specific time period. The period is configurable at the printer or in the printer's TFTP file (specified in the bootptab entry), but many users simply use the factory default, which is 90 seconds.

The timer only runs when the printer is available but the host is not sending data, as is the case during a mid-file suspension. The timer does not run when the printer is unable to print, i.e., it has been taken offline, or places itself offline due to a paper out or toner low condition.

The behavior of SUSPENDSPOOL differs from its functionality in release 2.05 and in earlier releases. This difference is a result of mapping the command onto the native mode spooler command set.

The difference permits RESUMESPOOL ... ;BACK nnn and RESUMESPOOL ... ;FORWARD nnn to function as they did in the 2.05 and earlier releases.

A consequence is that SUSPENDSPOOL does not relinquish ownership of a spool file that is printing.

To force the spooler process to relinquish ownership, you may choose either of two methods:

```
SUSPENDSPOOL 6
SPOOLER 6;RELEASE
```

or

```
SPOOLER 6;SUSPEND;NOW;NOKEEP
```

Suspending a class

To suspend spooling where class LP consists of logical devices 6, 9, and 11, and where their spooler processes retain ownership of any spool files that they are currently processing, enter:

```
SPOOLER LP ;SUSPEND
```

Suspending with an offset

You can specify an offset when you suspend spooling. Suppose that the spooler for logical device 6 is printing page 20. You want to suspend spooling so that when you direct the spooler to resume, it begins printing page 17. Enter:

```
SPOOLER 6 ;SUSPEND; KEEP; OFFSET = -3
```

The KEEP parameter directs logical device 6 to retain ownership of the spool file that it is currently processing. The KEEP option need not be entered. It is the default.

You can use the RELEASE parameter of the SPOOLER command to release a retained spool file.

Resuming Spooling

Use the NMS command SPOOLER with the parameter RESUME or the command RESUMESPOOL.

Both SPOOLER ... ;RESUME and RESUMESPOOL begin printing where the printer left off, provided that the following conditions are met:

- You suspended the spooler with either SUSPENDSPOOL (without the ;FINISH option) or SPOOLER *ldev* ;SUSPEND (without the ;FINISH option but with the ;KEEP option and with no specified offset).
- You did not enter SPOOLER *ldev* ;RELEASE while the spooler was suspended. (The RELEASE parameter directs a suspended spooler to release a spool file that it is currently retaining.)

Now suppose that logical device 6 is owned by a spooler process. Enter:

```
SPOOLER 6 ;SUSPEND
```

Suppose that the spooler had just transmitted the 20th line of page 10 and suppose that the conditions above hold. To continue spooling at the 21st line of page 10 as if the suspension never took place, enter:

```
SPOOLER 6 ;RESUME
```

If you do not suspend this way, the spooler prints a trailer and prints a header when it resumes. If the spooler releases a file, any specified offset is honored by the next spooler that prints the file. If there is no offset, the next spooler starts at the beginning of the page at which the previous output was suspended.

Displaying the Status of the Input/Output Devices

Use the NMS command SPOOLER with the parameter SHOW or the command SHOWDEV. This displays the status of the spooling process associated with the device(s) or class(es) specified. To display the status of all devices in class PP, enter:

```
SPOOLER PP;SHOW
```

The displays for SPOOLER ... ;SHOW and SHOWDEV look different but they display some of the same information. SPOOLER ... ;SHOW displays more information for spooling processes than SHOWDEV does.

- The SPOOLER ... ;SHOW field QSTATE matches the SHOWDEV field AVAIL.
- The SPOOLER ... ;SHOW field OWNERSHIP matches the SHOWDEV field OWNERSHIP.

If you use the SHOW parameter, the display shows the current state of the selected spooler(s) at the time the command executor completes processing the command. This means that the spooler(s) may be in a pending (have not completed requested operation) state rather than

a requested state. If this is true, an asterisk (*) precedes the state of the process on the display.

Listing Input and Output Spool Files

To display information about one or more spool files, use one of these:

- the command `LISTSPF`
- the commands `SHOWIN` and `SHOWOUT`

These produce a listing of input and output spool files and their characteristics.

The `LISTSPF` command provides more information than the `SHOWIN` and `SHOWOUT` commands and the display for `LISTSPF` is different from the `SHOWIN/SHOWOUT` display. The `LISTSPF` command display both input and output spool files.

With `LISTSPF`, you can request a particular group of spool files from the specified set by using the `SELEQ` parameter, you can specify more than the default information by using the `DETAIL` parameter, and you can display a summary by using the `STATUS` parameter.

The NMS assigns a unique SPOOLID number to each spool file. Only spool files get SPOOLID numbers. Spool files are no longer assigned DFID numbers. Only unspooled device files are assigned DFID numbers. DFID and SPOOLID numbers can have the same numeric value because they are kept in separate tables.

When you use `LISTSPF`, you see the SPOOLID number. This number ranges from 1 to 9,999,999. (A DFID number ranges from 1 to 32767).

To list your input and output spool files, enter:

```
LISTSPF @
```

To list spool file O2050 with a two line description, enter:

```
LISTSPF O2050 ;DETAIL
```

Note

If you use the `SHOWIN` or `SHOWOUT` commands, you may get both the SPOOLID and the DFID numbers depending on your parameters. See the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115) for details.

Copying Spool Files

To copy a spool file, use one of these:

- the COPY command
- the FCOPY command

You may copy a nonprivate spool file from OUT.HPSPOOL to your account with the COPY or FCOPY commands. The file copied with FCOPY or COPY is not linked to the spooling subsystem.

To copy spool file O2050 to an unlinked file called NEWFILE, enter:

```
FCOPY FROM=O2050.OUT.HPSPOOL; TO=NEWFILE;NEW
```

Displaying Spool File Content

To view the contents of a spool file, use one of these:

- the PRINT command
- the PRINTSPF utility
- the SPIFF utility
- the SPOOLF utility
- any text editor that edits variable-length files and that does not have restricted file codes. (EDIT/3000 is such an editor)
- the FCOPY utility

PRINTSPF displays a spool file's contents in a formatted manner so that you can examine the contents of both the data and the special overhead area of each record. You can display a portion of your spool file, the line numbers, and a certain number of characters on each line by using PRINTSPF.

Because spool files are ordinary disk files, you can display them using a text editor. The text editor eliminates the need for the SPOOK commands TEXT, LIST, and FIND. Spool files have variable-length records. You must use an editor that handles this. If you use EDIT/3000, enter the SET VARIABLE command before you copy a file into your workspace. Then the records are not truncated to 255 bytes.

To display output spool file O2050 using EDIT/3000, enter:

```
EDITOR
SET VARIABLE
SET LENGTH=lengthofrecord
SET RIGHT=lengthofrecord
T O2050.OUT.HPSPOOL
L ALL
```

The EDIT/3000 command `SET VARIABLE` ensures that the editor treats the file as variable-record. The commands `SET LENGTH` and `SET RIGHT` override the default display of 72 characters in EDIT/3000.

You may use EDIT/3000—or any editor that edits variable-length files and that does not have restricted file codes—to modify the contents of a spool file, but you cannot save the modified text under its current name in the `OUT.HPSPOOL` group. You may save it under any valid file name in your local file space. The new, modified file becomes unlinked.

To display output spool file O2345 23 lines at a time, using `PRINT`, enter:

```
PRINT O2345.OUT.HPSPOOL
```

To display lines 23 through 56 of spool file O2345 using `PRINTSPF`, enter:

```
PRINTSPF "FILE=O2345 ;START=23 ;END=56"
```

`PRINTSPF` searches the current logon directory for O2345 first. If the file is not found, `PRINTSPF` then searches `OUT.HPSPOOL`.

Renaming a Spool File

Use `RENAME`.

Since spool files are ordinary disk files, `RENAME` renames *unlinked* spool files if you have access to them. You cannot rename a spool file that is in `IN.HPSPOOL` or `OUT.HPSPOOL`. You must first copy the spool file to your account and then rename it. When you copy a spool file into your account, however, it is no longer linked to the spool file subsystem.

To rename output spool file O1234, which is in your account, enter:

```
RENAME O1234, new1234
```

Storing/Restoring Spool Files

You may store and restore spool files in NMS format or in SPOOK5/SPXFER format.

NMS format

To store or restore one or more spool files in NMS format to tape, use this combination:

- the STORE command.
- the RESTORE command

The STORE and RESTORE commands create and restore spool files *only* in the NMS format. Such spool files cannot be restored to MPE V/E systems.

SPOOK5/SPFXFER

To store and restore spool files in a SPOOK5 or SPFXFER format, use the SPXFER utility.

The CI STORE command accommodates the storing of nonprivate output spool files to tape. You may store your own nonprivate output spool files that are in OUT.HPSPOOL and that are linked to the spooling subsystem. You do not need system manager or account manager capabilities. You cannot store or restore input spool files because they are private files.

The RESTORE command accommodates the restoring of output spool files from tape. A spool file successfully restored to OUT.HPSPOOL is automatically linked to the spooling subsystem. The spool file's SPOOLID is changed to match the name assigned by the file management routines.

Note

Spool Files on a STORE/RESTORE tape cannot be restored to an MPE V/E or MPE/iX spooler systems that do not contain NMS. Spool files stored on tape with either the SPOOK or SPOOK5 utility cannot be restored with STORE/RESTORE.

To store spool file O23, enter:

```
FILE SPBACKUP ;DEV=TAPE
STORE O23.OUT.HPSPOOL ;*SPBACKUP ;SHOW
```

To store from indirect file INDFILE, an ASCII file that contains the following:

```
O234.OUT.HPSPOOL,O456.OUT.HPSPOOL;SHOW

STORE ^INDFILE; *T
```

Suppose that your spool files 012, 034, and 056 are stored to tape and suppose that you are USER, the creating user, and are in the group and account GROUP.ACCOUNT at store time. The spool files themselves reside in OUT.HPSPOOL. Now suppose that you want NEWUSER to be the creator of the spool files when you restore them.

Enter:

```
RESTORE *T ;@.OUT.HPSPOOL
;CREATOR=NEWUSER
```

The spool files are in OUT.HPSPOOL and NEWUSER.ACCOUNT is the creator of the spool files. The names of the spool files are changed to match the SPOOLID names assigned by the file management routines and are automatically linked to the spool file directory (SPFDIR).

Transporting Spool Files Between Types of Systems

The utility SPFXFER transfers spool files between MPE/iX systems containing NMS and MPE/iX systems not containing NMS or MPE V/E Systems. This utility transfers NMS spool files to tape in a format that MPE V/E SPOOK and MPE V/E SPOOK5 (version G.02.B0 and later) understand. The SPFXFER utility can read SPOOK tapes from any version of MPE/iX and MPE V/E. Chapter 2 contains more information about SPFXFER.

Replacing SPOOK

MPE/iX provides three utilities, SPIFF, PRINTSPF, and SPFXFER, for manipulating spool files. Consult Chapter 5 for details on these utilities.

Table B-1. SPOOK Replacements

SPOOK Command Replaced	Use MPE/iX Command or Utility
SHOW	LISTSPF or SPIFF
PURGE	PURGE, SPOOLF ... with DELETE parameter, or SPIFF
ALTER	SPOOLF with ALTER parameter, or SPIFF
INPUT	RESTORE or SPFXFER
OUTPUT	STORE or SPFXFER
TEXT, LIST, and FIND	An appropriate text editor or PRINT or PRINTSPF or SPIFF
MODE	PRINTSPF or SPIFF
COPY	SPIFF
APPEND	SPIFF

C

Page Level Recovery and Checkpoints

The native mode spooler (NMS) supports Page Level Recovery (PLR) and checkpoints, which provide the ability to restart printing at any page of the document due to device problems (such as power failure, paper jams, running out of paper, ink, toner, etc.) or a suspended spooler.

The Page Level Recovery and checkpoint functions recover spooler output in different ways. Which method you use is determined by the type of device you are using. Several devices support PLR, whereas only CIPER protocol printers (i.e., the HP 256x series with HP-IB interface) support checkpoints. The following sections describe PLR and checkpoints, and how to use them.

What is Page Level Recovery?

Page Level Recovery is best illustrated by two examples:

- Suppose the spooler is printing the 27th page of a spool file when the paper jams. After an operator has cleared the jam and repositioned the paper, recovery is the combined features of the spooler, the file system, and the printer which allows printing to resume automatically at the beginning of page 27 and continue from there.
- Suppose the spooler runs out of paper after printing the 27th page of a spool file. Recovery is the same set of features which allows printing to resume automatically at the beginning of page *28*, since page 27 was completely printed.

Whenever a spooler selects a spool file to print, it retrieves the final page printed the last time the file was selected. If this is the first copy of the file, or the previous copy printed to completion, the page is 0. This special value alerts the spooler to bypass recovery and start with the first page.

If this value is non-zero, the spooler sends a silent run control sequence to the printer. (Assume for this part of the discussion that the printer supports silent running). The sequence includes the target page value. The spooler begins sending spool file data in the normal fashion. As long as this data is for a page preceding the target page, the printer interprets it as if it were printing it, but does not actually print it on paper. When the target page is reached (as determined solely by the printer), the printer starts printing as well as interpreting.

The process is almost the same for a jam recovery or device power failure. After either of these situations occur and has been corrected, appropriate status is reported to the spooler. The spooler then asks the printer for the last page it successfully printed. That page + 1 becomes the new target page, and the spooler then invokes the same recovery mechanism described at the start of this section.

When is recovery used?

The spooler invokes the same mechanism in all the following situations:

- When the printer reports a device power failure or paper jam.
- When printing starts anywhere but at the beginning of the spool file:
  - A combination of `SUSPEND` and `RESUME` to any page other than 1.
  - Selecting a spool file whose output was interrupted the last time it was selected due to:
    - System shutdown,
    - A `SPOOLF` command (such as `;DEFER` or `;DEV=<newdev>`),
    - A `SPOOLER <ldev>;STOP` or `;SUSPEND;NOW;NOKEEP` command.

    In any of these cases, the final page printed is stored with the spool file. When the spool file is again selected for printing, printing starts with the page following the last one printed.

Note that recovery is unexpected in the first case, but is expected in all the other situations.

Kinds of Page Level Recovery

There are three varieties of Page Level Recovery:

- Restarting at a user-specified page number in a command such as `SPOOLER 6; RESUME; OFFSET=-5`.
- Restarting at the point of interruption when a device exception occurs (such as a device power failure).
- Restarting at the point of interruption when a particular spool file is (re)printed after being interrupted previously by a command such as `SPOOLF #O`*nnn*`; DEFER` or `SPOOLER 6; STOP`.

All three types of PLR use *silent run* techniques. Silent run is a mode in which data is sent to a printer and the printer interprets it but does not start actual printing until the printer reaches a target starting page specified by the host, in this case, the spooler.

Page Level Recovery requires a bidirectional interface to the printer and a specific Printer Job Language (PJL) feature support in the printer, namely, the `JOB` and `EOJ` commands, and the `PAGE` and `JOB` variables of the `USTATUS` command. For more information on PJL, refer to the *Printer Job Language Technical Reference Manual* (5961-0636).

Note

If either the bidirectional interface or required PJL support is missing, and printing is interrupted for any reason (device power failure, operator command, etc.), the entire file must be reprinted when printing is resumed. If the interruption is due to a device exception, a warning message to this effect is displayed on the system console or the $STDLIST of a user who has Associated the device. If it is due to an operator command, a similar message is displayed on the $STDLIST of the user who issued the command.

Components of successful Page Level Recovery

Recovery requires the *combined* features of the spooler, the file system, and the printer. The following subsections describe how each component contributes to recovery.

The spooler

Access to the entire spool file. The spooler manages the entire output object (the spool file). It is the only component of the three that can see the global picture. If the spooler were not present, meaning that the user application was sending output directly to the device, any recovery would have to be managed by the application.

This is often done by applications such as check printing programs, which print to unspooled devices using numbered checks whose supply is carefully controlled to prevent fraud. Such applications communicate with the printer operator to determine which check numbers are successfully printed and which are not, for any reason (paper jam, misalignment, ribbon malfunction or loss of ink, etc.).

Because the spooler can access any part of the spool file, it can select any part of the file as a starting point for retransmission. Its choice is determined by the other two components. This is a key feature of successful recovery.

Device independence. Another important property of the spooler is *device independence*. The spooler is a high level application that communicates with the printer. It does not know whether it is printing to an HP 2680 page printer, an HP 5000 page printer, an HP-IB CIPER printer, or a serially-connected printer. Any differences are managed at the printer storage manager (PSM) level of the file system. As a result, all communications to and from the file system and printer use the same interfaces and identical structures, regardless of which variety of printer may be connected to this particular spooler process. The spooler uses standard file system intrinsics such as `FWRITE` and `FDEVICECONTROL` for non-network printers.

This results in a “lowest common denominator” approach. This means that the spooler must support protocol demanded by any of the four printer types. If a particular protocol is not used or needed by the other types, its PSM must ignore it.

For example, at the end of every copy the spooler sends a special control sequence indicating that the copy has finished. The serial printer PSM uses this to disconnect the printer from either its DTC or its dialup line if no new spool file is started within a certain number of seconds. Since the spooler does not know whether its device is truly a serial printer, it must send this sequence after every copy. Non-serial printer PSMs simply ignore it (by reporting successful completion).

The file system

The file system supports spool file recovery in the following ways:

- An enhanced status reporting and control interface allows printer and PSM status to be reported to the spooler, and allows the spooler to tell the printer its proper restarting page.
- The PSMs hide printer differences from the spooler. All status reporting and spooler control uses a device-independent model. The model is based on the CIPER protocol, since that protocol has the richest feature set for recovery, but is used for non-CIPER printers as well.

The printer

The printer is an equal partner (with the spooler and the file system) in achieving reliable page-level recovery, that is, recovery to the start of a specific page. The printer contributes to recovery in two ways:

- Only the printer knows where pages truly start. There are a number of vertical motion control sequences:
  - With the printer set at *n* LPI, a <CR> advances paper 1/n inch.
  - A half-line feed, Esc=, advances paper 1/2n inch.
  - The %200 series of CCTL codes advances paper a given number of lines at *n* LPI without regard to page breaks.
  - The %300 series advances paper a given number of lines determined by Vertical Forms Control (VFC) configuration.
  - If end-of-line wrap is enabled, the printer can advance paper without an explicit vertical motion control.
  - Bar codes advance paper differently from all of the above.
  - Other sequences advance paper based on dot rows or decipoints.

  Different printers support different subsets of these control sequences. Some are configurable or purchasable options in the same printer. For the spooler to determine page breaks, it would need a complete print engine based on the actual printer used. It would also need to know the configuration *on the printer* of all options which affect vertical motion. This is totally impractical, and, from a performance standpoint, unacceptable.

- The printer's second contribution is the ability to *silent run*. This allows (re)printing to start at some page other than the beginning of the output without printing all the pages that preceded it.

Here is a list of the printers that support PLR:

- Color LaserJet
- LaserJet 4 family (except 4L)
- HP 5000/C30 and C40
- HP 2680
- HP 2688
- HP-IB CIPER

Checkpoints

A checkpoint is a snapshot of the state of a printer at a particular point in the data stream. If that point happens to be at the top of page N, *and* the native mode spooler can restore that state to the printer, it starts printing immediately at page N without having to silent run through pages 0 to N-1.

Checkpoints are a feature of the CIPER protocol, implemented only in the HP-IB version of HP256x printers. The HP-IB printers generate checkpoints at the top of each page.

How do checkpoints work?

The spooler sends data through the file system to the printer, and the printer begins to print it. At some point, the printer advances paper to the top of the second page. It generates a checkpoint, and notifies the spooler (through many layers of status reporting) that the checkpoint is available for retrieval. Whenever the spooler receives this status, it requests the checkpoint from the printer and saves it in the checkpoint file.

Note

Because of buffering in the data path, as well as the asynchronous nature of checkpoint reporting, the spooler may be well into data destined for page 2 (or even page 3 or later) of the report by the time it receives the checkpoint for the end of page 1.

Printing continues. Suppose at page 100 an operator enters `SPOOLER <ldev>; SUSPEND; OFFSET=-20`. Since the `;NOW` keyword is assumed by default, the spooler flushes all buffered data to the printer, sets the target page to 80, and suspends.

Later the operator enters `SPOOLER <ldev>; RESUME`. Using the recovery process described earlier, it fetches the target page (80), searches the checkpoint file for the corresponding checkpoint, downloads the checkpoint to the printer, then begins sending data from (near) that point in the spool file. The printer silent runs for a short period until the proper data for page 80 arrives, then it begins to print. Note that it was not necessary for the spooler to send the first 79 pages.

Checkpoint files

Although only CIPER protocol devices return checkpoints, the spooler always creates a checkpoint file. This is primarily for device independent operation. The checkpoint file always has a dummy "checkpoint 0" entry. When silent running is invoked, the spooler first looks for the closest checkpoint preceding the target page as a starting point for silent running. The checkpoint 0 entry assures that such a checkpoint always exists. If it is used, the silent run starting point is the beginning of the spool file, as expected.

The spooler creates a unique checkpoint file for each device which prints a given spool file for the following reasons:

- Different devices result in different files. As noted earlier, CIPER protocol devices return checkpoints, other devices do not. A checkpoint file generated as a result of printing to a CIPER protocol device would not be usable if another copy was printed to an HP 2680.
- Even among similar devices, the state information in a checkpoint may vary. If an existing checkpoint file entry does not exactly match that of the entry for the same page returned from the printer, the spooler marks the checkpoint file as corrupt. This is discussed in more detail in the "Reality check" section later in this appendix.

Usable for additional copies

The spooler generates a checkpoint file as it prints the first copy of a spool file. Once generated, the file can be used when printing subsequent copies. A later example in the "No checkpoints while silent running" section, shows how a forward search (page 100 in the example) requirs a lengthy silent run cycle. If, instead, the first copy had printed without interruption, it would be possible to start a second or subsequent copy (on the same printer) at page 100 with a minimal silent run delay. This is because the checkpoint for page 100 was generated as the first copy was printed, and can now be used to go directly to page 100.

A checkpoint file is deleted with its spool file

Because the checkpoint file is created automatically by a spooler process, it is also deleted automatically when the corresponding spool file is deleted, either following its final copy, or when deleted by a command (`SPOOLF` or `PURGE`). The checkpoint file is also deleted if the spool file moves to the SPSAVE state.

If more than one checkpoint file exists for the same spool file, because the spool file was printed on more than one device, they are all deleted.

Occasionally a checkpoint file remains after its spool file has been deleted. Prior to MPE/iX Release C.50.00, this usually occurred because a spool file was being stored to a tape when a spooler process tried to delete it. The Store process deleted neither the

spool file nor the checkpoint file when it finished storing them. This problem has been corrected in MPE/iX Release C.50.00.

Any user with SM capability can purge any checkpoint file which is not being accessed. Even if the corresponding spool file still exists, no harm is done. If the file is selected for printing, the spooler generates a new checkpoint file. If the file is deleted without being printed, the system does not care if no checkpoint file exists.

File naming convention

The spooler generates one checkpoint file per spool file for each device which prints any part of the spool file. A checkpoint file is named C*nnnnn*, where *nnnnn* is the numerical value of the spoolid of the corresponding spool file. The checkpoint filename is placed in device name groups in the `HPSPOOL` account.

There are two ways to name a checkpoint file:

- By the corresponding spool file and the device name

  For example, suppose LP1 is a device name, and the native mode spooler creates the output spool file `O1234.OUT.HPSPOOL` that is printed by device LP1. Then the name of the companion checkpoint file is `C1234.LP1.HPSPOOL`.

- By the spool file and the logical device number

  Whenever a spooler process is started for a device, a group is created in the HPSPOOL account (if it does not already exist). This group is named D*mmmmmmm*, where the *mmmmmmm* represents the logical device number (with leading zeros as required) of the device.

  For example, if a spooler process is started for LDEV 6, the corresponding group in HPSPOOL is D0000006. If #O1234 is selected for printing by LDEV 6's spooler process, the spooler creates checkpoint file C1234.D0000006.HPSPOOL (unless it already exists from an earlier selection by the same spooler process). If #O1234 is later routed to LDEV 19, LDEV 19's spooler creates C1234.D0000019.HPSPOOL if necessary.

Checkpoint file space

Checkpoints are not new with the native mode spooler. They were introduced on MPE V/E when the HP-IB CIPER printers were first released. Their generation is strictly a printer phenomenon, but the spooler catalogs them and uses them in a recovery situation.

On MPE V/E, the catalog of checkpoints was in the spool file's user labels. The number of user labels available, and the size of the checkpoint structure limited the number of checkpoints to 32. These were managed in a circular fashion, with the 33rd checkpoint replacing the first, and so on. The spooler stored every eighth checkpoint (remember, the printer generates one at the top of each page), allowing a range of 256 pages. This is why the `RESUMESPOOL` command had a 256 page limitation, and why successive iterations of the command to exceed 256 pages did not work.

The native mode spooler removed the arbitrary 256 page limitation by cataloging checkpoints in a separate file. This file theoretically can grow to the maximum 4Gb, but that would never happen, as any reasonable spool file contains much more data per page than a checkpoint entry. Therefore, a hypothetically very large spool file would hit the 4Gb limit long before its checkpoint file did.

In practice, the initial allocation of checkpoint file space is quite small. In addition, non-CIPER printers do not generate checkpoints. Therefore, the file does not grow beyond its initial allocation, so disk space for checkpoint files is usually not a problem. Even though checkpoint files are used for non-CIPER printers (because they are managed by the spooler, which must be device independent), such files contain only a few structural entries.

Corrupted checkpoint files

As mentioned earlier, whenever the spooler for a particular CIPER protocol device prints the same page more than once, there exists a cataloged checkpoint in the checkpoint file, and a checkpoint which is returned by the device. The spooler enforces an exact match between these two. If it detects any difference, the checkpoint file is marked corrupted. Subsequent checkpoints are not cataloged, nor are any earlier ones reused. Note that any recovery situation then requires silent running from the beginning of the spool file.

The only way to re-establish the use of the checkpoint file is to start a new copy of the spool file at the beginning. This logically deletes any earlier contents and marks the checkpoint file as usable. Any checkpoints received during this copy are then cataloged normally.

Why are such stringent conditions enforced? Because the same spool file printed on the same device should yield the same checkpoints. If it does not, the checkpoint file data is suspect and should not be used. The only disadvantage is that any recovery situation requires more time than if a valid checkpoint file were available. But this approach also assures that no error occurs as a result. The recovery will succeed, it just takes longer.

Checkpoints versus silent running

Note the difference between checkpoints and silent running. The HP 2680, HP 5000, color LaserJet, LaserJet 4 family, and all CIPER printers (HP-IB versions of HP 256x printers) all support silent running. Only the CIPER printers support checkpoints. Checkpoints make it possible to minimize the time a printer spends silent running.

Serially-connected printers do not have hardware support of silent running. The serial printer storage manager emulates silent running for a subset of possible conditions. Recall that full emulation would require a complete device-specific print engine to account for all possible vertical motion control sequences.

For the purpose of supporting silent running, the serial PSM assumes 66 lines per physical page, with top and bottom margins of three lines each. It also uses the HP standard default VFC specification. A user-specified VFC (using FDEVICECONTROL 64) is not supported. Within those limitations, it recognizes and interprets the full range of carriage control codes, in both prespace and postspace modes. However, it does not scan data looking for PCL or other vertical motion sequences.

Checkpoint considerations

Must be CIPER printer

Only CIPER protocol printers support checkpoints. The only CIPER printers are the HP 256x series with HP-IB interface. Why do other printers not provide checkpoints? Because checkpoints are a snapshot of a printer state, and the large page printers such as the HP 2680 and HP 5000 simply have too much state information. The font table (and all the downloaded bitmapped fonts), the forms overlay table, the logical page table are all part of the printer state. It would require an extremely large checkpoint file to save all this information for each page printed. It would also take substantial link bandwidth to upload the data from the printer, thereby subtracting from overall printer performance.

The HP 2680, HP 5000, and Color LaserJet, and LaserJet 4 family printers do not return checkpoints, but *do* support silent run. This means that in a recovery situation it is not necessary to reprint the entire file, but it is necessary for the spooler to transmit the file from its beginning (with the printer in silent run mode until the target page is reached). In our earlier example, this corresponds to silent running through the first 79 pages of data instead of going directly to page 80. The reason for this is that the beginning of the file is the only place where the spooler and the printer agree on the printer's state. Without checkpoints, there is no way for the spooler to know the printer's state as of a given page. Therefore, there is no way to restore that state in a recovery situation.

Non-recoverable checkpoints

Even with a CIPER printer, a checkpoint is only valid when it represents the full state of the printer as of a given page. Such a checkpoint is a *recoverable checkpoint*, but "recoverable" is usually omitted from the description. If the state of the printer cannot be fully represented in the space available in the checkpoint structure, the checkpoint is called a *nonrecoverable checkpoint*, meaning it cannot be used in a recovery situation.

Downloading a Vertical Forms Control (VFC) image causes all succeeding checkpoints to be nonrecoverable. It is the only user option that causes nonrecoverable checkpoints. Since VFC images are usually downloaded at the beginning of a spool file, this means that the printer is forced to silent run from the beginning of the spool file in any recovery situation.

No checkpoints while silent running

The printer does not generate checkpoints when it is in silent run mode. This can be significant, as the following example shows:

Suppose the printer prints the first 20 pages of the first copy of a spool file. An operator then suspends and resumes the spooler at page 100. Because there is no checkpoint for page 100, the printer must silent run from page 21 to page 100, then resume printing. It does so, and at page 120 the operator again suspends spooling and resumes it at page 80. Because the printer did not generate checkpoints while it was silent running through pages 21-99, the closest checkpoint not exceeding page 80 is that for page 20. It is downloaded, but the printer must then silent run from there until page 80.

Checkpoint file must be logically consistent

Whenever the spooler retrieves a checkpoint from the printer, it checks whether or not that checkpoint has already been cataloged in the checkpoint file. If so, the cataloged checkpoint must identically match the one returned by the printer. If it does not, the checkpoint file is considered corrupted. The spooler does not use it unless a new copy of the spool file is started from the beginning, thereby regenerating the entire checkpoint file.

This phenomenon can lead to long silent running, even for spool files that previously generated recoverable checkpoints. Without a checkpoint close to the target page, the spooler is forced into silent running from the beginning of the spool file.

Checkpoints defined by printer's physical top-of-form

As mentioned earlier, CIPER printers generate a checkpoint at the top of each page. But there are two kinds of pages. The physical page length is settable only on the printer's front panel (that is, it cannot be set programmatically, and so is unavailable even to the spooler). Logical page lengths can be set through the proper PCL command sequence, but are usually set by the VFC definition in the CIPER ENV file specification. For example, `VFC,6,66` defines a logical page length of eleven inches (66 lines at 6 lines per inch).

The printer generates a checkpoint each time it reaches the top of a *physical* page. If the physical page length is different from the logical page length, you do not get expected results in a recovery situation.

Printer must stop at physical top-of-form

For the printer to generate a checkpoint, paper must stop on the line representing the physical top-of-form. Many applications position paper in the most efficient manner by advancing from the final printed line on one form to the first printed line on the next. If this first line is not at the top of the physical form, the printer does not generate a checkpoint.

D

Migration Information and Limitations

The native mode spooler (NMS), MPE/iX version B.40.00 and later versions, is a complete replacement of spooling subsystems on previous MPE V/E and MPE XL systems. With MPE/iX version B.40.00 and beyond, SPOOK and SPOOK5 (MPE V/E) functions are replaced by the SPIFF utility. The SPOOK utility is still supported for all MPE V/E releases and for all MPE XL releases that contain it.

With MPE/iX version B.40.00, SPOOK functions are carried out with SPIFF commands and with CI commands, editor subsystems, the SPFXFER utility, the PRINTSPF utility, STORE, and RESTORE.

With the NMS, linked spool files are created as variable-length disk files and are kept in a special account named HPSPOOL. HPSPOOL handles spool files and spooler processes and contains two special groups named IN and OUT. Input spool files reside in the IN.HPSPOOL group and output spool files reside in the OUT.HPSPOOL group.

Unlinked output spool files are also disk files and can reside in any group and account.

Caution

Nothing but NMS spool files should be in the IN and OUT groups of the HPSPOOL account. Do not purge the IN and OUT groups. Also, HPSPOOL contains checkpoint file groups. Do not tamper with these groups.

An input spool file is a private file (level 2 privileged file). The system creates an input spool file by the STREAM command or by a spooler process controlling a spooled input device.

An output spool file is either a private or nonprivate file that is linked or unlinked. The system creates linked output spool files by using HPFOPEN or FOPEN to open a spooled device. Unlinked output spool files can be created by using HPFOPEN or the BUILD command.

The FILE and BUILD commands have an option called SPOOL that specifies an output spool file not linked to the spool file directory (SPFDIR). This file does not print nor is it linked to the SPFDIR. When you issue the SPOOLF command with the PRINT option, a linked copy of the spool file is made. Refer to chapter 2 for an example.

See chapters 1 and 2 for more details about input, output, and private spool files.

Note

If you use COPY or FCOPY to copy a spool file from the HPSPOOL account, the copy file is an *unlinked* spool file.

The COPY command of the SPIFF utility copies a linked spool file to a new linked spool file.

A separate spooler process manages each spooled device. A logical device number or a device name, which is an alphanumeric string of up to eight characters, defines a device. In addition, a device class defines a related set of devices. You may issue a single command for a device class, and it affects all devices that belong to that class.

Setting Up Some Basic Tasks

Configuring devices

Some extensions to device configuration apply to spooling. Refer to the *System Startup, Configuration, and Shutdown Reference Manual* (32650-90042) for information about configuring a device.

Spool File space and limits

With NMS in place, you need not use SYSGEN to configure MAX NUMBER OF SPOOLFILE KILOSECTORS or the maximum number of open spool files. Instead, you set a file space limit for the HPSPOOL account with the ALTACCT command. You can set a file space limit on the IN and OUT groups independently with the ALTGROUP command if necessary. During installation or the initial update, the HPSPOOL account and its groups are created with unlimited file space.

Each spool file is potentially 4 Gbytes long, the same as most files on MPE/iX.

Allowing users control of spooled devices with the ASSOCIATE facility

If you are a general user, you must use the ASSOCIATE command to gain control of a device class. This command links a device class, such as LP, to an individual user on the system. Before you can be associated, the system manager must run a utility program (the version of ASOCTBL.PUB.SYS that matches your operating system) in order to create a device class user association table. This table defines which users may be associated with which device classes. Multiple users can be in the table for a given class, but only one user at a time is allowed to be the controller of the devices in a device class at any given time.

The following operator commands that relate to spooling are then available to you once you are associated to a device or if you are at the console:

```
SPOOLER dev ;START
            ;STOP
            ;SUSPEND
            ;RESUME
            ;OPENQ
            ;SHUTQ
ALTSPOOLFILE
DELETESPOOLFILE

FORMSALIGN

HEADOFF
HEADON
OPENQ
OUTFENCE
RESUMESPOOL
SHUTQ
STARTSPOOL
STOPSPOOL
SUSPENDSPOOL
```

dev is either a logical device number, a device class, or a device name. More information about the `ASSOCIATE` command is in the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-60115).

Initiating spooling

To initiate spooling, you can use the `SPOOLER` command with the `START` parameter as follows:

```
SPOOLER dev ;START
```

The *dev* parameter is a logical device number, a device class, or a device name.

You can also use the `STARTSPOOL` command. The commands `SPOOLER` *deviceclass* `;START` and `STARTSPOOL` *deviceclass* affect all devices that belong to that device class just as though you issued the `SPOOLER ... ;START` or `STARTSPOOL` command for each individual device. Previously, `STARTSPOOL` *deviceclass* only opened spooling queues for that device class. Any `FOPEN` using that device class name would generate a spool file. Spooling processes were not started for the actual devices in the device class.

Automatically initiating spooling with system startups

There are two methods that you can use to automatically spool devices each time you boot the system. They include using the `SYSSTART` file or configuring devices as initially spooled.

The system startup file (SYSSTART.PUB.SYS) may contain commands to enable spooling and to start spooling processes. What follows is an example of a system startup file.

```
STARTUP
ALLOW @@.@@;COMMANDS=LOG
comment System Startup File
OUTFENCE 14
spooler 6;openq
spooler 19;openq
spooler 18;start
streams 10
headoff 18
limit 5,30
outfence 6
jobfence 7
Welcome SYSMSG.MESSAGE
VMOUNT ON,AUTO
Comment End of systart file.
```

The system manager oversees any changes to the system startup file.

You can also use SYSGEN to configure devices so that they are automatically spooled during system startup. Refer to the *System Startup, Configuration, and Shutdown Reference Manual* (32650-90042).

Migrating

The NMS is part of the fundamental operating system and is installed on the MPE/iX system with the INSTALL or UPDATE utility. The first boot after an INSTALL or an UPDATE creates the HPSPOOL account structure.

With release A.40.00, device classes are now treated solely as collections of logical devices. This means that operations applied to a device class are applied to all devices in a class. For example, if a spooler is stopped for a device class, the spooler is also stopped for all logical devices in that class.

Before release A.40.00, users could issue the following command:

```
STOPSPOOL ldev
```

and print "hot" to the printer while spool files were being created for the device classes associated with that LDEV. This is no longer possible since device classes are collections of logical devices.

There are two workarounds shown in the examples below. If the device that you want to operate unspooled is part of a class that contains at least one other device that can remain spooled, there is no problem. The following examples assume that this other device does not presently exist.

- Method 1: Configure at least two devices in a particular device class.

  Suppose that LDEV 6 and nonexistent LDEV 19 are configured for device class LP. Make sure that the queue for LDEV 19 is open. One way to do this is to enter an `OPENQ 19` command in your `SYSSTART` file. If LDEV 6 is usually spooled, you can also enter the `STARTSPOOL 6` command or `SPOOLER 6;START` command in your `SYSSTART` file.

  Make sure that your applications generating spooled output direct that output to class LP. Enter:

  ```
  STOPSPOOL 6
  ```

  or

  ```
  SPOOLER 6;STOP
  ```

  Start your application. When it is finished, enter:

  ```
  STARTSPOOL 6
  ```

  or

  ```
  SPOOLER 6;START
  ```

  to print the accumulated spool files.

  Output directed to class LP creates spool files even while LDEV 6 is unspooled because the queue is open for at least one device (LDEV 19) in the class.

- Method 2: This workaround requires no dummy device.

  Enter:

  ```
  STOPSPOOL 6

  SPOOLER 6;STOP
  ```

  Start the application requiring unspooled access to the printer. Then enter:

  ```
  OPENQ 6
  ```

The first method is recommended. In the second method, there is a period of time during which any process—even one that directs its output to class LP—can acquire the printer unspooled. If it is a process other than the intended one, you must either wait for that process to close the printer, or you must abort it.

This situation does not arise with the first method because there is always at least one device in class LP with its queues open.

There are two reasons for device classes being treated as collections of logical devices:

- Confusion. Many users were confused by the distinction between queues and spooler processes. More confusion arose because of the

different ways some of the commands operated on a logical device and the device class to which the logical device belonged.

- Simplification. Previously, managing separate queues for logical devices and their device classes was messy. You could never be sure when you would get a spool file or a hot device.

Finding SPOOK in job streams

Since the SPOOK utility is obsolete in MPE/iX, it is important to locate references to SPOOK in job streams and to replace them with the appropriate commands (or the appropriate NMS commands instead. You can use an editor to do this.

Spool File transport

You may transport output spool files between MPE/iX systems through STORE and RESTORE. Since input spool files are private files, you cannot write them to tape with STORE. Also, you cannot store private output spool files. You can find more information about STORE and RESTORE in chapter 2.

The utility SPFXFER transfers files to tape in a format that MPE V/E SPOOK and MPE V/E SPOOK5 (with the exception of MPE V/E releases prior to G.02.B0) can read. For releases prior to G.02.B0, SPOOK cannot read SPFXFER tapes. The SPFXFER utility can read tapes created by SPOOK from any release of MPE/iX, MPE XL, MPE V/E, and SPFXFER itself. The SPFXFER utility has four commands that are similar to the commands of the SPOOK utility. For more information about SPFXFER, see chapter 2.

Recovery

The NMS uses a checkpoint file, described briefly in chapter 2 and in more detail in appendix D, for recovery. The checkpoint file saves page checkpoints and other data which are used to recover printing following an interrupt such as a power failure or the execution of the SPOOLER ... ;RESUME or RESUMESPOOL commands.

Device recovery

- If the device returns page checkpoints (for example, CIPER protocol devices), the spooler recovers to a specific page.
- The HP 2680, HP 2688, and HP 5000 series printers do not support page checkpoints, but performs a silent run from the beginning of the spool file to the page where the interruption occurred.
- For CIPER devices and for HP 2680, HP 2688, and HP 5000 series printers, the output resumes printing at the correct page.
- Since serial printers cannot recover accurately to a specific page, they may not resume printing at the correct page or they may shift the page boundary from its original position.
- The NMS retains output spool files until the device notifies the spooler that physical output is complete and error free.

Spool File recovery

Since spool files are permanent disk files, recovery is no more complicated than it is for other disk files.

- When the system executes the INSTALL command, all files, including spool files, are purged. The system preserves all *output* spool files for all other startups, including UPDATE.
- The system only recovers input spool files for every START with the RECOVERY option. There is a job master table (JMAT) entry for each job input spool file. The JMAT is only recovered during a START RECOVERY.
- The system rebuilds the spool file directories for all such startups *except* INSTALL from information kept with each spool file.
- Output spool files that are in the CREATE state when the system is interrupted may not be completely recovered. Any data not posted to the disk before the system interruption cannot be recovered. If no data was posted, the spool file is deleted by the recovery mechanism.
- After a boot, if no output spool files are recovered, the next SPOOLID assigned is O1. If any spool files are recovered, the next SPOOLID counter is one greater than the largest SPOOLID of the recovered spool files.
- If the system is booted with the NORECOVERY option of the START command, an apostrophe (') is inserted between the J or S and the number in each spool file's associated job or session number. For example J1234 becomes J'1234. This is to indicate that the job or session number was not generated from the current job or session counters.

Spool File states

Spool File states include the states ACTIVE, READY, OPEN, CREATE, PRINT, DEFER, PROBLM, XFER, DELPND, and SPSAVE. The LISTSPF command displays these states. Figure C-1 shows a one-to-one correspondence between the LISTSPF states and the SHOWIN states for input spool files. Figure C-2 shows a one to one correspondence between the LISTSPF states and the SHOWOUT states for output spool files.

Table D-1. Input Spool File States

LISTSPF	SHOWIN
OPEN	OPEN
ACTIVE	ACTIVE
READY	READY

Table D-2. Output Spool File States

LISTSPF	SHOWOUT
READY	READY or LOCKED
CREATE	OPEN
DEFER	READY, D
DELPND	(underlying states are READY or ACTIVE)
SPSAVE	READY, D
PROBLM	READY, D
PRINT	ACTIVE
XFER	ACTIVE or LOCKED

The LISTSPF command, described in chapter 4, contains information about the states.

DEFER state

DEFER is a state of an output spool file, not just the condition of a spool file at priority 0 as in the past. A deferred spool file retains its original output priority but does not print, even if the priority is above the outfence. This feature allows users to see the original output priorities of deferred spool files. The spool file keeps this priority if its state changes to READY.

If you use the commands SPOOLF ... ;DEFER or SPOOLF ... ;UNDEFER on a file in the CREATE or PRINT states, the state that the command LISTSPF displays does not change until the file is closed either by the user process (CREATE) or the spooler process (PRINT). At that time, the file enters the DEFER or READY state depending on which command processed last.

Note

A SPOOLF ... ;DEFER command issued to a file in the PRINT state causes the NMS to interrupt the spooler process currently printing the file. The spooler ends its print job and returns the file to NMS file management.

XFER state

The XFER state indicates that a spool file has been selected for transportation from one node of network to another. It may be displayed and used as a STATE in a selection equation. It is provided for use as desired by third-party software providers. The spooler never places a file in the XFER state nor uses the state as a basis for spooler actions.

The DFID and the SPOOLID

Before version A.40.00 of MPE XL, the spooler kept track of individual spool files by assigning a unique device file identifier (DFID) number to each file. The MPE/iX (and MPE XL) NMS assigns a unique spool file identification number (SPOOLID) to each spool file. SPOOLIDs are similar to DFID, but are not the same. The IDD and ODD (CM tables) management routines assign DFIDs to unspooled input and output device files such as terminals, tape drives, and unspooled printers. The NMS assigns SPOOLIDs only to spool files. The input and output spool file directory (`SPFDIR`) keeps SPOOLIDs. Since they are kept in different tables, a DFID and a SPOOLID may have the same numeric value.

Similarities between a DFID and a SPOOLID include their format (#O*nnnn* or #I*nnnn*) and the replacement of a SPOOLID number for a DFID number in the `ALTSPOOLFILE` and `DELETESPOOLFILE` commands.

For the `FFILEINFO` intrinsic, item number 38 requests a 16-bit SPOOLID. SPOOLIDs can be up to 9,999,999 and only SPOOLIDs up to 32,767 fit in the 16-bit representation. A new `FFILEINFO` item number, 78, which returns the same information as item number 38 has been added to accommodate SPOOLIDs larger than 32,767. If the SPOOLID fits in the 16 bit representation (less than 32,676), it is returned; otherwise, a zero (0) is returned.

Control information display

Use the `SPIFF` or `PRINTSPF` utility to view control information such as page eject locations and double spacing. Information about `SPIFF` and `PRINTSPF` is in chapter 5.

Outfence

The system outfence default is 14.

Error handling

All non-CI error messages are in `SYSCAT.PUB.SYS`, the native mode message catalog. You can access them by using the MPE/iX error management procedures and intrinsics. All CI messages for MPE/iX version A.40.00 and beyond are in Set 2 of `CATALOG.PUB.SYS`, the compatibility mode message catalog.

The status of spooler error messages are numbered according to the information part of the error status. You can call the MPE/iX error management intrinsics either to print the error or to put it into a buffer for further manipulation.

SPOOK Limitations

The SPOOK utility does not exist on MPE/iX version A.40.00 and later versions. Command interpreter (CI) commands, the PRINTSPF utility, the SPFXFER and SPIFF utilities, STORE, RESTORE, editor subsystems, and the FCOPY subsystem, replace SPOOK commands. Refer to Appendix B and chapter 2.

Device Limitations

Output devices

The native mode spooler (NMS) supports the following output devices:

- HP 2680, HP 2688, and HP 5000 series laser page printers
- the HP 256x CIPER protocol printers
- serially connected printers such as the LaserJet series and the HP 293x series
- serially connected plotters

The NMS does not support the HP 2608S.

Input devices

A tape drive is the only device supported for input spooling.

Device classes

With release 2.1, device classes are treated solely as collections of logical devices. This means that operations applied to a device class are applied to all devices in a class. For example, if a spooler is stopped for a device class, that spooler is stopped for all logical devices in that class.

There are two reasons for device classes being treated as collections of logical devices:

- Confusion. Many users have been confused by the distinction between queues and spooler processes. More confusion arose because of the different ways some of the commands operated on a logical device and the device class to which the logical device belonged.
- Simplification. In earlier releases, managing separate queues for logical devices and their device classes was messy. You could never be sure when you would get a spool file or a hot device.

Before release 2.1, users could issue the following command:

```
STOPSPOOL ldev
```

and print "hot" to the printer while spool files were being created for the device classes associated with that LDEV. This is no longer possible since device classes are collections of logical devices.

There are two workarounds. If the device that you want to operate unspooled is part of a class that contains at least one other device that can remain spooled, there is no problem. The following examples assume that this other device does not presently exist.

- Configure at least two devices in a particular device class. Suppose that LDEV 6 and nonexistent LDEV 19 are configured for device class `LP`. Make sure that the queue for LDEV 19 is open. One way to do this is to enter an `OPENQ 19` command in your `SYSSTART` file. If LDEV 6 is usually spooled, you can also enter the `STARTSPOOL 6` command or `SPOOLER 6;START` command in your `SYSSTART` file.

  Make sure that your applications generating spooled output direct that output to class `LP`. Enter:

  ```
  STOPSPOOL 6
  ```

  or

  ```
  SPOOLER 6;STOP
  ```

  Start your application. When it is finished, enter:

  ```
  STARTSPOOL 6
  ```

  or

  ```
  SPOOLER 6;START
  ```

  to print the accumulated spool files.

  Output directed to class LP creates spool files even while LDEV 6 is unspooled because the queue is open for at least one device (LDEV 19) in the class.

- This workaround requires no dummy device. Enter:

  ```
  STOPSPOOL 6
  ```

  or

  ```
  SPOOLER 6;STOP
  ```

  Start the application requiring unspooled access to the printer. Then enter:

  ```
  OPENQ 6
  ```

The first method is recommended. In the second method, there is a period of time during which any process—even one that directs its output to class LP—can acquire the printer unspooled. If it is a process other than the intended one, you must either wait for that process to close the printer, or you must abort it.

This situation does not arise with the first method because there is always at least one device in class `LP` with its queues open.

Recovery limitations

Device recovery

The NMS uses all available device features to aid in device recovery, either automatically (following a device power failure) or manually triggered by an operator.

- The CIPER devices support page checkpoints. The spooler uses the checkpoints to recover quickly to a specific page. Refer to appendix D for detailed information on checkpoints.
- The HP 2680, HP 2688, and HP 5000 series perform Page Level Recovery (PLR), which uses a silent run technique from the beginning of the file to the point of recovery. The devices's knowledge of pages ensures that output resumes correctly. Refer to appendix D for detailed information on Page Level Recovery.
- A serial printer has no feedback to tell the spooler of its page location. Output may not resume at the correct page boundary, or the page boundary may be shifted.

A full discussion of paging exists in the description of the `SPOOLER ... ;RESUME` command in chapter 4.

Spool file recovery

- The system preserves output spool files for all system startups *except* `INSTALL`.
- Job input spool files have a one-to-one correspondence with job master table (JMAT) entries. The JMAT is still a CM table, so it is still rebuilt for updates and `START NORECOVERY`. Whenever an update or `START NORECOVERY` occurs, the system purges all input spool files.
- The NMS retains output spool files until the device positively notifies the spooler that paper output is in the output station and error free.

See appendix C for more information about recovery.

Space limitations

Since spool files are permanent disk files, their size limit is 4 gigabytes. This is currently the file size limit of any MPE/iX disk file.

You may want to limit the linked spool file space on the `HPSPOOL` account and on its groups `IN` and `OUT` by using the `ALTACCT` and `ALTGROUP` commands, respectively. If you do this and encounter a directory file space limit on `HPSPOOL` or its `IN` or `OUT` groups at the time of the creation of a linked spool file, all spooling queues are globally disabled. A message is displayed on the system console stating that all spooling queues are disabled because of an account or group file space limit. You must use the `OPENQ @@` command when the condition is resolved. The message is as follows:

```
ALL SPOOLING QUEUES HAVE BEEN GLOBALLY
DISABLED DUE TO A FILE SPACE LIMIT ON
THE HPSPOOL ACCOUNT OR ITS GROUPS. USE
THE OPENQ @@ COMMAND TO GLOBALLY ENABLE
THE SPOOLING QUEUES WHEN THE
CONDITION HAS BEEN CORRECTED.
```

You may also run out of system domain disk space available to spool files. If this happens, all spooling queues are globally disabled and a similar message to the previous one is displayed. The message is as follows:

```
ALL SPOOLING QUEUES HAVE BEEN GLOBALLY
DISABLED DUE TO LACK OF SYSTEM DISK
SPACE. USE THE OPENQ @@ COMMAND TO
GLOBALLY ENABLE THE SPOOLING QUEUES
WHEN THE CONDITION HAS BEEN CORRECTED.
```

Editor limitations

You can browse output spool files using any editor that supports variable-length record files. EDIT/3000 is one editor that you can use. To use EDIT/3000 to browse a spool file, you must use the command `SET VARIABLE` *before* copying the spool file into your work space. Suppose you want to browse spool file `O2050.OUT.HPSPOOL`. Enter:

```
editor

set variable
set length=NN
set right=NN
t o2050.out.hpspool
```

NN is the record length of the longest record in the file. *NN* must be <= 255

Now you may use any of the EDIT/3000 commands.

Note

The editor should only be used to browse the spool file.

The system does not let you overwrite the original spool file in `OUT.HPSPOOL`. You can save the spool file in text file in a group to which you have access, but the file you save is no longer a linked spool file. In fact, it is no longer a spool file.

You may not browse input spool files.

Deleting a spool file, limitations

When you delete a linked spool file with the `PURGE` command, the system removes the spool file immediately because the `PURGE` command opens the file with exclusive access.

This is different from what happens when you use `SPOOLF ... ;DELETE`. If a spool file is not open, not printing, or not being stored, `SPOOLF ... ;DELETE` removes the spool file immediately. Otherwise, it is put into the `DELPND` state. The system notifies any spooler process printing the spool file, and printing stops at that point. Then the system deletes the spool file when its last user (except the `STORE` program) closes the file.

The `DELETE` parameter of the `SPOOLF` command works for `DATA` input spool files in the `READY` state. It also works for all output spool files not in the `CREATE` state. It *does not* work for job `$STDIN` spool files. You must use the `ABORTJOB` command to delete job `$STDIN` spool files.

For more information, refer to the `SPOOLF` command with the `DELETE` parameter in chapter 4.

Note

If the system cannot gain exclusive access to the file (for example, if a spooler is printing the file), `PURGE` fails.

The system deletes unlinked spool files in the same manner as it deletes any permanent disk files.

Renaming a spool file, limitations

You may not rename spool files linked to the spooler queues. @

Glossary

banners — The generic term for either the header or trailer of a printout. It contains identification information for the listing.

batch job — A batch job is the noninteractive execution of a series of MPE/iX commands and/or user programs. The commands and programs are preceded by a valid `JOB` command and followed by the `EOJ` command. Batch jobs are submitted to the system with a spooled input device or the `STREAM` command.

checkpoint — A snapshot of the state of a printer at a point in the output known to both the device and the spooler. By using checkpoints in the appropriate device commands, the spooler can quickly reestablish the state of a device as of the time of the checkpoint. A CIPER protocol printer generates a checkpoint at the top of each page.

checkpoint file — A small file that the spooling subsystem creates and manages. The spooler keeps checkpoints returned from the device, as well as other data that it needs to recover properly from printing interruptions. There is one checkpoint file per spool file per device on which the spool file is printed. All checkpoint files for a given spool file are deleted when the spool file is deleted.

CIPER — An acronym for control of intelligent peripherals. CIPER is a spooler printer cooperative protocol designed to foster rapid recovery from a device interruption such as a power failure. A device using this protocol generates checkpoints and returns them to the spooler, which saves them in a checkpoint file. Printer output may be interrupted by either a device failure or a user command. When output resumes, an appropriate checkpoint is retrieved from the checkpoint file and is downloaded to the printer, restoring the state of the printer to that checkpoint. In this way, it is possible to start transmitting spool file data at points other than the beginning of the file. For large output spool files, the time saved is quite noticeable. Currently, the only peripherals that support CIPER protocol are the HP-IB connected HP 256x family of line printers.

CM spooler — CM spooler refers to the compatibility mode spooler released with all versions of MPE/iX before release A.40.00.

conditional top-of-page — The motion of the logical pen or physical paper such that the next output begins at the top of the next logical page, on the same physical sheet, or on a new physical sheet (depending on use). *Conditional* means that this motion does not occur if the the pen is already at the top of the page due to an explicit `FOPEN` or `FCLOSE` of a spooled device file by a user, or a record with a carriage-control character of one (octal 61).

data file — An input spool file that has been entered through a device or streamed using the `DATA` command. T he data in it will be used later by an interactive session or batch job.

FLABX — An acronym for file label extension. Each MPE file has a label in which are stored attributes common to all files (such as its unique file identifier, or UFID). The FLABX is an optional additional area associated with the file label in which information may be stored that is not part of the data in the file. For spool files, the NMS stores attributes such as file state (`READY`, `PRINT`), output priority, and number of copies in the FLABX.

linked spool file — A spool file that has an entry in the `SPFDIR` and, therefore, is known to the spooling subsystem. A linked spool file is always in the reserved account `HPSPOOL`. Input spool files are in `IN.HPSPOOL` and output spool files are in `OUT.HPSPOOL`. Only linked output spool files can be scheduled for printing by a spooler. Linked input spool files are used by a CI. If you copy a spool file from `OUT.HPSPOOL` to your group and account, that copy has no `SPFDIR` entry and is therefore not a linked spool file.

MOM — `MOM` is a child process of `PROGEN`, the master system process. `MOM` creates all other spooling processes.

operator — The person who monitors the system console and manages the computer on a daily basis. This includes establishing job and session limits, setting the output fence, responding to users' resource requests, loading the system after a shutdown or failure, and informing users of the system's status. Operator is also called console operator or system operator.

private spool file — A spool file that is created with the `PRIVATE` option specified. The private option is intended for applications that produce sensitive output. Private spool files have more stringent access and attribute restrictions than nonprivate spool files.

selection equation — A method of selecting one or more spool files from a larger group of spool files according to user-specified criteria. The selection equation is not an independent command or intrinsic. It is a feature of the `LISTSPF` and `SPOOLF` commands. Further details may be found in the description of the `LISTSPF` and `SPOOLF` commands in chapter 4.

silent run A method of recovery to a particular page following an interruption of the printing process. The interruption can be expected, as in `SPOOLER ... ;SUSPEND`, or unexpected, as in a device power failure. Silent run requires hardware support in the device or software support in the device's storage manager. When in silent run mode, the device or storage manager interprets, but does not print, all data sent to it. When it reaches the page at which it should start printing, it does so automatically and without additional spooler control. Some devices, such as the HP 2680, must silent run from the beginning of the spool file to the restart point. Others, such as CIPER devices, are capable of silent running from a location closer to the desired start point. Serial printing devices do not support any form of silent run, so any silent running must be simulated by the device's storage manager or the device must restart at the beginning of the file.

SPFDIR Spool File directory. There are two `SPFDIR`s, one for input spoofiles and another for output spool files. Each `SPFDIR` is an internal table used by the native mode spooler to keep information about spool files that are linked (known to the spooling subsystem). Attributes such as target device, output priority, and number of copies are kept in an `SPFDIR` entry. Each `SPFDIR` contains a working copy of this information, built from the master copy (kept in each spool file's FLABX) when the system is booted and whenever new linked spool files are created.

SPIT Spooling process information table. This is an internal table used by the native mode spooler to keep information about spooling processes. Attributes such as process state (`ACTIVE`, `IDLE`, `SUSPEND`) and current SPOOLID (if any) are kept in the SPIT entry.

spool Acronym for simultaneous peripheral operation online. A facility that permits concurrent usage of devices that would otherwise be nonshareable, such as tape drives and printers. This is accomplished by copying the input from or output to these devices to disk, where it waits until the required process (input) or device is available. The operation is called *spooling*, and the program that accomplishes it is called a *spooler*. This facility includes commands for monitoring and controlling the spooled devices and the spooled files on disk.

spooler A process that manages input from or output to nonshareable devices so that they appear to be shared among several users. The input spooler collects data from an input device (usually a tape drive) and places it in a disk file for later use by a CI or user process. The output spooler oversees the orderly selection and printing of spool files.

spool file

The term spool file refers to a file originating from or directed to a nondisk spooled device. When a nonshareable device is spooled, any user program attempting to access the device is actually accessing an opened input spool file or a created output spool file instead of the device itself. Associated with each spool file (except `DATA` files) is a job or session number, a file designator, a user name, an account name, a device name, the state of the file, and a SPOOLID. (A `DATA` file may, but need not, have a file designator. It has no job or session number until it is opened by the user. It has all of the other attributes listed.) Spool Files may be in one of the following states: `OPEN`, `ACTIVE` (input spool files only), `READY`, `DELPND` (input or output spool files), `CREATE`, `PRINT`, `DEFER`, `PROBLM`, `SPSAVE`, or `XFER` (output spool files only). These states describe different steps in the life of a spool file.

SPOOLID

The NMS equivalent of the CM spooler device file ID (DFID). The primary difference is that it can range from 1 to 9,999,999. The SPOOLID is the number that follows the #O's or #I's in the `LISTSPF` display and the spool file portion of the `SHOWIN` or `SHOWOUT` display. It is assigned by the NMS file management routines when the spool file is first created and is associated with the file for its entire lifetime. It determines the filename of the spool file.

storage manager

The lowest level of the three MPE/iX file system abstractions. The storage manager is responsible for resolving all device specific requirements into a common interface for higher levels. For example, all printers support the concept of a device job. All data is printed between the start and end of a device job, but different printers have different ways of being told to start and end a job, and different responses to these commands. The storage managers accept a generic `START DEVICE JOB` command and issue whatever unique device commands are required to implement the generic command.

stream

A concept similar to input spooling by which users submit batch jobs to MPE/iX. An input spooler is a system process controlling a device that reads batch job record images into an input spool file for later execution. The `STREAM` command runs in a user process and accesses a file of batch job record images, reading these images into an input spool file for later execution.

system manager

The person who manages the computer installation, who is responsible for creating accounts, and who defines the resource use limits and capabilities for each user.

type manager

The middle level of the three MPE/iX file system abstractions. The type manager is a filter for file access methods. For example, a tape type manager would allow the intrinsic `FREADBACKWARD`, where a disk type manager would return an error. For a second example, a disk type manager for RIO files would allow the intrinsic `FDELETE`, while the disk type manager for non-RIO files would not.

Index