

Spooler Utilities Reference Manual

Abstract

This manual describes the Spooler utilities—Peruse, Spoolcom, Font, and RPSetup—and presents the complete syntax for these utilities. It also presents a general introduction to the Spooler subsystem. This manual is intended for all interactive and operational users of the Spooler subsystem.

Product Version

Spooler D48, H01 and H02

Supported Releases

This manual supports D48.03, G06.16 and H06.03 releases and all subsequent releases until otherwise indicated in a new edition.

Part Number	Published
522295-003	July 2012

Document History

Part Number	Product Version	Published
135721	Spooler D41	August 1997
140947	Spooler D41	May 1998
422816-001	Spooler D46	June 1999
425735-001	Spooler D46	February 2000
522295-001	Spooler D48	November 2001
522295-002	Spooler D48	May 2002
522295-003	Spooler D48, H01 and H02	July 2012

Legal Notices

© Copyright 2012 Hewlett-Packard Development Company, L.P.

Legal Notice

Confidential computer software. Valid license from HP required for possession, use or copying.
Consistent with FAR 12.211 and 12.212, Commercial

Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Export of the information contained in this publication may require authorization from the U.S. Department of Commerce.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

Intel, Pentium, and Celeron are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

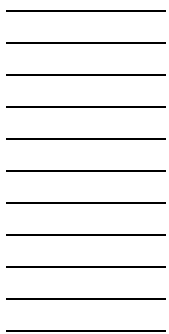
Motif, OSF/1, UNIX, X/Open, and the "X" device are registered trademarks, and IT DialTone and The Open Group are trademarks of The Open Group in the U.S. and other countries.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc. OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE OSF MATERIAL PROVIDED HEREIN, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. OSF shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.

© 1990, 1991, 1992, 1993 Open Software Foundation, Inc. The OSF documentation and the OSF software to which it relates are derived in part from materials supplied by the following: © 1987, 1988, 1989 Carnegie-Mellon University. © 1989, 1990, 1991 Digital Equipment Corporation.

© 1985, 1988, 1989, 1990 Encore Computer Corporation. © 1988 Free Software Foundation, Inc. © 1987, 1988, 1989, 1990, 1991

Hewlett-Packard Company. © 1985, 1987, 1988, 1989, 1990, 1991, 1992 International Business Machines Corporation. © 1988, 1989 Massachusetts Institute of Technology. © 1988, 1989, 1990 Mentat Inc. © 1988 Microsoft Corporation. © 1987, 1988, 1989, 1990, 1991, 1992 SecureWare, Inc. © 1990, 1991 Siemens Nixdorf Informationssysteme AG. © 1986, 1989, 1996, 1997 Sun Microsystems, Inc. © 1989, 1990, 1991 Transarc Corporation. OSF software and documentation are based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. OSF acknowledges the following individuals and institutions for their role in its development: Kenneth C.R.C. Arnold, Gregory S. Couch, Conrad C. Huang, Ed James, Symmetric Computer Systems, Robert Elz. © 1980, 1981, 1982, 1983, 1985, 1986, 1987, 1988, 1989 Regents of the University of California.



Spooler Utilities Reference Manual

Index	Examples	Figures	Tables
-----------------------	--------------------------	-------------------------	------------------------

[Legal Notices](#)

What's New in This Manual	ix
Manual Information	ix
New and Changed Information	ix
About This Manual	xi
Who Should Use This Manual	xi
How This Manual Is Organized	xi
Related Manuals	xii
Your Comments Invited	xiii
Notation Conventions	xiii

[1. Spooler Quick Start](#)

Command Files	1-1
Spooler Configuration File	1-2
Spooler Start Macro	1-2
Starting the Spooler	1-4
Warm Starting a Spooler	1-4
Cold Starting a Spooler	1-5
Cold Starting a Drained Spooler	1-8
Draining the Spooler	1-9
Recovery for the Spooler	1-12
Recovery for an Improperly Drained Spooler	1-12
Rebuild the Spooler Control Files and the Job Bit Map	1-12
Connecting Spoolers in a Network Using RPSsetup	1-13

[2. Introduction to the Spooler Subsystem](#)

Spooler and Spooler Plus Comparison	2-2
Spooler Features	2-2
Why Use the Spooler?	2-2
Spooler Components	2-3
Collectors	2-5

- [Spooling in a Network](#) 2-7
- [Peruse and Spoolcom Comparison](#) 2-9

2. Introduction to the Spooler Subsystem (continued)

- [Spooler Jobs and Job Attributes](#) 2-9
 - [Batch Number](#) 2-9
 - [Job Priority](#) 2-10
 - [Job Copies](#) 2-10
 - [Job Report Name](#) 2-10
 - [Job Form Name](#) 2-10
 - [Job Location](#) 2-10
 - [Job State](#) 2-10
- [Devices and Device Attributes](#) 2-12
 - [Device Form Name](#) 2-12
 - [Device Header Message](#) 2-12
 - [Device State](#) 2-14
 - [Selection Algorithm](#) 2-14
- [Routing Structure](#) 2-15
 - [Broadcast and Nonbroadcast Groups](#) 2-16
 - [Default Routing](#) 2-16
 - [Implicit Route Creation](#) 2-17
- [Using the Spooler](#) 2-17
 - [Sending Process Output to a Spooler Location](#) 2-17
 - [Sending Process Output to a SPOOL DEFINE](#) 2-18
- [The Spooler and Batch Jobs](#) 2-18
 - [How to Change the Attribute Values of Batch Jobs](#) 2-21
 - [Linking and Unlinking Jobs](#) 2-21
 - [Ensuring Device Uniqueness for Batch Jobs](#) 2-21
- [Spooler Security](#) 2-22
 - [Status Requests](#) 2-22
 - [All Other Requests](#) 2-22

3. Peruse

- [Peruse Features](#) 3-2
- [Running Peruse](#) 3-3
 - [Running Peruse Interactively](#) 3-4
 - [Running Peruse Noninteractively](#) 3-4
 - [Initial Job Status Display](#) 3-5
 - [Entering Peruse Commands](#) 3-6
 - [Declaring the Current Job](#) 3-7

- [Displaying a Job](#) 3-8
- [Using Spooler Job Files](#) 3-8

3. Peruse (continued)

- [Peruse Commands](#) 3-9
- [BATCH Command](#) 3-11
- [COPIES Command](#) 3-12
- [DEL Command](#) 3-13
- [DEV Command](#) 3-15
- [EXIT Command](#) 3-16
- [FC Command](#) 3-17
- [FILES Command](#) 3-18
- [FIND Command](#) 3-19
- [FORM Command](#) 3-21
- [HELP Command](#) 3-23
- [HOLD Command](#) 3-24
- [HOLDAFTER Command](#) 3-26
- [JOB Command](#) 3-28
- [LINK Command](#) 3-32
- [LIST Command](#) 3-33
- [LOC Command](#) 3-38
- [NUMCOL Command](#) 3-39
- [OPEN Command](#) 3-40
- [OWNER Command](#) 3-41
- [PAGE Command](#) 3-42
- [PRI Command](#) 3-44
- [PURGE Command](#) 3-45
- [REPORT Command](#) 3-46
- [SHOWGROUP Command](#) 3-47
- [SJFILES Command](#) 3-49
- [STARTCOL Command](#) 3-50
- [STATUS Command](#) 3-51
- [UNLINK Command](#) 3-53
- [VOLUME Command](#) 3-54

4. Spoolcom

Spoolcom Features	4-2
Spoolcom Security	4-2
Running Spoolcom	4-2
Running Spoolcom Noninteractively	4-3
Running Spoolcom Interactively	4-3
Entering Spoolcom Commands	4-4
Spoolcom Commands	4-5
BATCH Command	4-7
COLLECT Command	4-12
COMMENT Command	4-19
DEV Command	4-20
EXIT Command	4-40
FC Command	4-41
FONT Command	4-42
HELP Command	4-47
JOB Command	4-48
LOC Command	4-58
OPEN Command	4-62
PRINT Command	4-63
SPOOLER Command	4-70

5. Font Utility

How the Font Utility Works	5-2
Running the Font Utility	5-3
Script File Syntax	5-5
Script File Skeleton	5-5
General Syntax Rules for Script File Directives	5-6
START-FONT and END-FONT Directives	5-6
FONT Directive	5-7
Normal Font Data	5-7
CONTROL Directive	5-9
CONTROLBUF Directive	5-9
SETMODE Directive	5-13
SOURCE Directive	5-14
START-LOCATIONS and END-LOCATIONS Directives	5-15
Comments in the Script File	5-16
Script File Example	5-17

6. RPSetup Utility

- [How the RPSetup Utility Works](#) 6-2
- [Interspooler Threads](#) 6-3
- [Running RPSetup](#) 6-6
- [Requirements for Running RPSetup](#) 6-7
- [Example](#) 6-8
- [Configuring Entities in the Local Spooler](#) 6-8
- [Default Configuration](#) 6-8
- [ASSIGN and PARAM Commands](#) 6-11

A. SPOOL Program

- [Running the SPOOL Program](#) A-1
- [EMS Collector DEFINE](#) A-4

B. Peruse Messages

C. Spoolcom Messages

D. Font Utility Messages

E. RPSetup Utility Messages

F. Using FASTP

- [Configuring FASTP](#) F-1
- [Tracing FASTP](#) F-2
- [Creating Trace Files](#) F-2
- [Specifying Trace Files for a Device](#) F-3
- [Enabling Tracing](#) F-3
- [Reading Trace Output](#) F-4
- [Extended Configuration With FASTCNFG](#) F-5
- [Creation](#) F-6
- [Redirection](#) F-6
- [Security](#) F-6
- [Alteration](#) F-6
- [Syntax](#) F-8
- [FASTCNFG File Redirection](#) F-10
- [EOLWRITE](#) F-12
- [ERRORRECOVERY](#) F-12
- [Considerations](#) F-12
- [TIMEOUT](#) F-12

G. Spooler Limits

H. Migration Considerations for Pre-D41 Spooler Versions

<u>Pre-D41 Interoperability Considerations</u>	H-1
<u>Pre-D41 Spoolcom and the D41, D46,D48 Spooler</u>	H-1
<u>Pre-D41 Peruse, D41, and the D46 and D48 Spooler Products</u>	H-1
<u>Pre-D41 Spooler, D41, and D46 and 48 Spoolcom Utilities</u>	H-1
<u>New Control File</u>	H-2
<u>Migrating Files From a Pre-D41 Product Version of Spooler</u>	H-2
<u>D41,D46, D48 Common Enhancements and Changes</u>	H-4
<u>Increased Object and Resource Limits</u>	H-4
<u>New Collector File Requirements</u>	H-4
<u>New Collector Data File Capacity Requirements</u>	H-5
<u>Improved Collector Performance</u>	H-5
<u>New Collector Data File UNIT and DP2 BUFFERSIZE Settings</u>	H-5
<u>D48 New Product Version of Spooler</u>	H-6
<u>Older-Version Collector Files Required to Access Older-Version Jobs</u>	H-6
<u>More Collector Openers for Each Collector</u>	H-7
<u>Warmstart From Older-Version Spooler Files</u>	H-7
<u>Changes to Spoolcom Commands</u>	H-7
<u>Changes to Spooler Library Procedures and Structures</u>	H-10
<u>Unsupported Procedures</u>	H-11
<u>New Versions of Print Processes Required</u>	H-11
<u>Increased Extended Memory</u>	H-11
<u>New EMS Event Messages</u>	H-11
<u>Increased Support for Failure Data System (TFDS)</u>	H-11
<u>Reduced Support for PSPOOL and PSPOOLB Print Processes</u>	H-12
<u>Extended Memory Changes for D46 and D48 Versions of the Spooler</u>	H-12

Index

Examples

Example 1-1.	Spooler Configuration File	1-2
Example 1-2.	Spooler Start Macro	1-3
Example 1-3.	Spooler Warm Start File	1-5
Example 1-4.	Spooler Cold Start File (COLDFILE)	1-6
Example 1-5.	Spooler Configuration File Named SPLCONF	1-7
Example 1-6.	FUP INFO Command To Find Spooler Control Files	1-10
Example 1-7.	PID of the Supervisor Process	1-11
Example 1-8.	Using the STATUS Command	1-11
Example 1-9.	Finding the Collector Files	1-11
Example H-1.	JOB Command to Status All Jobs in All Datafiles	H-8
Example H-2.	JOB Command to Status all Jobs in a Particular Datafile	H-9
Example H-3.	JOB Command to Status a Particular Datafile and Location	H-9
Example H-4.	Transferring Jobs	H-10

Figures

Figure 2-1.	Components of the Spooler	2-4
Figure 2-2.	Linked Spoolers in a Network	2-8
Figure 2-3.	Life Cycle of a Spooler Job	2-11
Figure 2-4.	Sample Header Page Produced by the FASTP Print Process	2-13
Figure 2-5.	The Spooler Routing Structure	2-15
Figure 2-6.	How the Spooler Determines Which Jobs Are to Be Batched	2-20
Figure 5-1.	Font Utility Operation	5-2
Figure 6-1.	Configuration Functions of the RPSetup Utility	6-2
Figure 6-2.	Interspooleser Threads	6-4

Tables

Table 2-1.	Collector Attributes	2-5
Table 2-2.	Key Attributes	2-19
Table 3-1.	Number of Lines Listed by the Function Keys	3-8
Table 3-2.	Summary of Peruse Commands	3-9
Table 4-1.	Spoolcom Command Summary for All Users	4-5
Table 4-2.	Spoolcom Command Summary for Authorized Users	4-6
Table 4-3.	Batch Job States During Which BATCH Subcommands Can Be Used	4-9
Table 4-4.	COLLECT STATUS Display Entries	4-16
Table 4-5.	Retryable Error Numbers	4-28
Table 4-6.	Device States During Which DEV Subcommands Can Be Used	4-31
Table 4-7.	DEV STATUS Display Entries	4-35

Tables (continued)

Table 4-8.	Typical Configuration for Laser Printer Type 5577	4-39
Table 4-9.	Job States During Which JOB Subcommands Can Be Used	4-52
Table 4-10.	JOB STATUS Display Entries	4-52
Table 4-11.	JOB STATUS Display Entries—Non-Detailed Single Job Status	4-53
Table 4-12.	JOB STATUS Display Entries—Detailed Single Job Status	4-54
Table 4-13.	LOC STATUS Display Entries	4-61
Table 4-14.	PRINT STATUS Display Entries (Without DETAIL Option)	4-67
Table 4-15.	PRINT STATUS Display Entries (With DETAIL Option)	4-68
Table 4-16.	SPOOLER STATISTICS Display Entries	4-72
Table 4-17.	SPOOLER STATUS Display Entries	4-74
Table 6-1.	ASSIGN and PARAM Command Default Values	6-8
Table 6-2.	Default Attributes Set by FASTP on Interspooled Transferred Print Jobs	6-13
Table F-1.	Types of Trace File Events and Data	F-4
Table G-1.	Spooler Limits	G-1
Table H-1.	D40 Spooler and D41 Spooler Limits Comparison	H-4
Table H-2.	Recommended BUFFERSIZE Settings	H-6

What's New in This Manual

Manual Information

Abstract

This manual describes the Spooler utilities—Peruse, Spoolcom, Font, and RPSetup—and presents the complete syntax for these utilities. It also presents a general introduction to the Spooler subsystem. This manual is intended for all interactive and operational users of the Spooler subsystem.

Product Version

Spooler D48, H01 and H02

Supported Releases

This manual supports D48.03, G06.16 and H06.03 releases and all subsequent releases until otherwise indicated in a new edition.

Part Number	Published
522295-003	July 2012

Document History

Part Number	Product Version	Published
135721	Spooler D41	August 1997
140947	Spooler D41	May 1998
422816-001	Spooler D46	June 1999
425735-001	Spooler D46	February 2000
522295-001	Spooler D48	November 2001
522295-002	Spooler D48	May 2002
522295-003	Spooler D48, H01 and H02	July 2012

New and Changed Information

Changes in the 522295-003 (July 2012) manual:

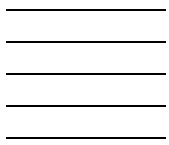
- Updated the D48 product template details [D48 New Product Version of Spooler](#) on page H-6.
- Added a note about COLLECT subcommand on page [H-10](#).

- Added a description about the limitation of the device name usage under the [Considerations](#) on page 4-33.
- Added a Peruse error message on page [B-8](#).

Changes in the May 2002 manual:

The following changes have been made to this manual:

- In [Section 3, Peruse](#), a new feature has been added to the LIST command. In previous releases of spooler, a job that contained more than 99,999 lines listed to an EDIT file would become truncated. Spooler now accommodates a maximum of 99,999,999 lines in an EDIT file. See the [LIST Command](#) for more information.
- In [Appendix H, Migration Considerations for Pre-D41 Spooler Versions](#), a new subsection has been added, [Migrating Files From a Pre-D41 Product Version of Spooler](#) on page H-2.
- Minor corrections have been made throughout the manual.



About This Manual

Purpose of This Manual

This manual describes:

- The Spooler utilities—Peruse, Spoolcom, Font, and RPSetup—and provides the command syntax and message descriptions for these utilities.
- The FASTP configuration process and gives the programmatic limits of Spooler components.

Who Should Use This Manual

This manual is intended for any users of the Spooler subsystem who wants complete command syntax information for the Spooler utilities. Users who want information about performing routine spooler operations should refer to the *Guardian System Operations Guide* and the *Guardian User's Guide*. Users who want to use the Spooler interface procedures to spool data programmatically should refer to the *Spooler Programmer's Guide*.

How This Manual Is Organized

The table below summarizes the contents of this manual.

Section or Appendix	Title	Contents
1	Spooler Quick Start	Provides step-by-step procedures for backing up, draining, coldstarting, and warmstarting the Spooler system.
2	Introduction to the Spooler Subsystem	Provides an overview of the Spooler subsystem
3	Peruse	Provides the syntax of all Peruse commands.
4	Spoolcom	Provides the syntax of all Spoolcom commands.
5	Font Utility	Describes the utility functions, gives the syntax of the script file for input to the utility, and gives the syntax of the command to run the Font utility.
6	RPSetup Utility	Describes the utility functions and gives the syntax of all RPSetup options.
A	SPOOL Program	Provides the syntax for running the Spool program.
B	Peruse Messages	Explains error and warning messages issued by Peruse and recommends ways to recover from errors.

Section or Appendix	Title	Contents
C	Spoolcom Messages	Explains error and warning messages issued by Spoolcom and recommends ways to recover from errors.
D	Font Utility Messages	Explains error and warning messages issued by the Font utility and recommends ways to recover from errors.
E	RPSetup Utility Messages	Explains error and warning messages issued by the RPSetup utility and recommends ways to recover from errors.
F	Using FASTP	Describes how to configure and use the FASTP print process.
G	Spooler Limits	Lists programmatic limits of spooler components.
H	Migration Considerations for Pre-D41 Spooler Versions	Describes migration considerations for the D41, D46 and D48 Spooler subsystems.

Related Manuals

The following manuals provide information about the Spooler utilities described in this manual:

- *Guardian User's Guide*

This guide provides a general introduction to the spooler, shows you how to use the Peruse utility to manage print jobs, and describes the non-restrictive Spoolcom commands. This guide is written for beginning users.

- *Guardian Operations Reference Summary*

This manual provides syntax summaries for the Font, Peruse, RPSetup, and Spoolcom utilities and the SPOOL program.

- *Spooler FASTP Network Print Processes Manual*

In G06.08, this manual replaced the TCP/IP LAN Print Spooler (T8932) document previously available on the site update tape.

- *The Spooler Programmer's Guide*

This guide describes the Spooler subsystem, its uses, and its applications for experienced programmers.

Notation Conventions

General Syntax Notation

The following list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS. Uppercase letters indicate keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example:

```
MAXATTACH
```

lowercase italic letters. Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

```
file-name
```

[] Brackets. Brackets enclose optional syntax items. For example:

```
TERM [\system-name.]$terminal-name
```

```
INT[ERRUPTS]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list may be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
LIGHTS [ ON           ]
        [ OFF         ]
        [ SMOOTH [ num ] ]
```

```
K [ X | D ] address-1
```

{ } Braces. A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list may be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name }
```

```
ALLOWSU { ON | OFF }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

... **Ellipsis.** An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address-1 [ , new-value ]...
```

```
[ - ] {0|1|2|3|4|5|6|7|8|9}...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

Punctuation. Parentheses, commas, semicolons, and other symbols not previously described must be entered as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate that the symbol is a required character that you must enter as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing. Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In the following example, there are no spaces permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing. If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] CONTROLLER
```

```
    [ , attribute-spec ]...
```

!i and !o. In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT ( segment-id           !i
                        , error                 !o
                        ) ;
```

!i,o. In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;           !i,o
```

!i:i. In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ ( filename1:length      !i:i
                           , filename2:length ) ;    !i:i
```

!o:i. In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ ( filename      !i
                       , [ filename:maxlen ] ) ;    !o:i
```

Notation for Messages

The following list summarizes the notation conventions for the presentation of displayed messages in this manual.

Nonitalic text. Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

Backup Up.

lowercase italic letters. Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

p-register
process-name

[] Brackets. Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list might be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
LDEV ldev [ CU %ccu | CU %... ] UP [ (cpu,chan,%ctlr,%unit) ]
```

{ } Braces. A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list might be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LBU { X | Y } POWER FAIL
```

```
process-name State changed from old-objstate to objstate
{ Operator Request. }
{ Unknown.           }
```

| **Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% **Percent Sign.** A percent sign precedes a number that is not in decimal notation. The %o notation precedes an octal number. The %b notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
```

```
P=%p-register E=%e-register
```

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

This requirement is described under [Backup DAM Volumes and Physical Disk Drives](#) on page 3-2.

Change Bar Notation

Change bars are used to indicate substantive differences between this edition of the manual and the preceding edition. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).

The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

HP Encourages Your Comments

HP encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

docsfeedback@hp.com

Include the document title, part number, and any comment, error found, or suggestion for improvement that you have concerning this document.

1 Spooler Quick Start

This section is a concise set of instructions for starting and draining, or stopping, the spooler.

Depending on the release from which you are upgrading, some configuration of spooler might be required. The best method for assuring that any previous version (T9101) of spooler can be used is to always back up the control files and the collector files for that version, prior to warm starting the new version of spooler. See [Appendix H, Migration Considerations for Pre-D41 Spooler Versions](#) for information on migrating previous versions of spooler.

Command Files

You should use command files to automate your spooler-related tasks. Spooler command files can have from one to 8 alphanumeric characters with the first character alphabetic. There are no default command-file names. Some of the command file names used in the examples of this section are defined as follows:

Command File Name	Purpose and Use
SPLCOLD	Cold starts the spooler and purges all jobs in the current collector. If your system does not have a \$SPLS process running, you can type OBEY SPLCOLD to start a basic spooler system.
SPLCONF	Configures the spooler. Is used by SPLCOLD.
SPLWARM	Warm starts the spooler.
SPLSTOP	Drains the spooler.
SPLDRAIN	

The default name of the spooler collector program is \$SYSTEM.SYSTEM.CSPOOL. Replace this file if you want to run your own collector process. For information on naming spooler components and files, refer to the *Guardian User's Guide*.

Spooler Configuration File

[Example 1-1](#) is a sample spooler configuration file:

Example 1-1. Spooler Configuration File

```

COMMENT*****
COMMENT  \ACCT $SPLS Spooler config file
COMMENT
COMMENT  Author;   John Scott
COMMENT  Date;     12/23/99
COMMENT  Function; To configure $SPLS supervisor
COMMENT
COMMENT*****

COLLECT  $S, FILE $SYSTEM.SYSTEM.CSPOOL
COLLECT  $S, DATA $OPER.SPLS.DATAS, UNIT 4
COLLECT  $S, CPU 0, BACKUP 1, PRI 148

PRINT    $SPLP, FILE $SYSTEM.SYSTEM.PSPOOL,DEBUG OFF
PRINT    $SPLP, CPU 1,BACKUP 0, PRI 147

DEV  $LPA,PROCESS $SPLP,SPEED 100 ,HEADER ON,WIDTH -1,TRUNC OFF
DEV  $LPA,RESTART OFF,TIMEOUT 360,EXCLUSIVE OFF,FIFO OFF,FORM,RETRY 5

DEV  \SOCL1.$S,PROCESS $SPLP,SPEED 100 ,HEADER OFF,WIDTH -1
DEV  \SOCL1.$S,RETRY 5,TIMEOUT 360,FIFO,TRUNC OFF,PARM -1

LOC  #LP.DEFAULT,DEV $LP.#A
LOC  #DEFAULT.DEFAULT,DEV $LP.#A
LOC  #LP5.DEFAULT, DEV \SOCL1.$S
LOC  #LP9.DEFAULT, DEV \SOCL1.$S
LOC  #MKTMIS2.DEFAULT, DEV \SOCL1.$S

SPOOLER, START

```

Spooler Start Macro

[Example 1-2](#) shows an example of a TACL macro that can be used to start the spooler:

Example 1-2. Spooler Start Macro (page 1 of 2)

```
?TACL MACRO
#set #informat tacl
==*****
==  \ACCT $SPLS SPOOLER START MACRO
==
== AUTHOR;      John L. Scott
== DATE;        2/24/97
== FUNCTION;    To start spooler $SPLS and collector $S.
== FILE;        $system.startup.strtspls
==
== To Cold Start; RUN $SYSTEM.STARTUP.STRTSPLS COLD
==  (Spooler isn't configured,to create new Spooler environment)
==
== To Warm Start; RUN $SYSTEM.STARTUP.STRTSPLS
==  (Spooler is configured and has been running prior to this coldload)
==*****

[#IF [#PROCESSEXISTS $SPLS]
| THEN | #OUTPUT *** WARNING *** PROCESS $SPLS ALREADY RUNNING
| ELSE |
  #PUSH #DEFAULTS

[#IF [#MATCH %1% COLD]
| THEN | == Make SURE it's a COLDSTART
  [#IF [#MATCH Y [#INPUT Are Your SURE You Want A COLDSTART (Y/N) >>] ]
| THEN |

== Set up security of OBJECT files
FUP PURGEDATA $OPER.SPOOLER.LOGFILE
FUP GIVE ($SYSTEM.SYSTEM.CSPOOL,$SYSTEM.SYSTEM.SPOOL),30,1
FUP SECURE ($SYSTEM.SYSTEM.CSPOOL,$SYSTEM.SYSTEM.SPOOL),nunu,progid

== Get rid of the old DATA COLLECTOR/SUPERVISOR FILES.
FUP PURGE ! $OPER.SPLS.*

== Create the $$S COLLECTOR file
FUP CREATE $OPER.SPLS.DATAS,EXT (1000,1000)
FUP GIVE $OPER.SPLS.DATAS,30,1
FUP SECURE $OPER.SPLS.DATAS,"NUNU"

== Start it UP
SPOOL /IN $OPER.SPLS.SPL,NAME $SPLS,PRI 150,NOWAIT, &
TERM $YMIOP.#CNSL,CPU 0/1,4095,4095,255, 15,15, 255, 4095
SPOOLCOM/IN $SYSTEM.STARTUP.SPLSCONF,OUT $OPER.SPOOLER.LOGFILE/

] == ENDIF
```

Example 1-2. Spooler Start Macro (page 2 of 2)

```

| ELSE | == It's a WARMSTART

FUP GIVE ($SYSTEM.SYSTEM.CSPOOL, $SYSTEM.SYSTEM.PSPOOL,    &
          $SYSTEM.SYSTEM.SPOOL), 30, 1

FUP SECURE ($SYSTEM.SYSTEM.CSPOOL, $SYSTEM.SYSTEM.PSPOOL,    &
            $SYSTEM.SYSTEM.SPOOL), "NUNU", PROGID

SPOOL /IN $OPER.SPLS.SPL, TERM $YMIOP.#CNSL, NAME $SPLS, NOWAIT, PRI 150, CPU 0/1

SPOOLCOM ;SPOOLER, START

] == ENDIF

#POP #DEFAULTS
#DELAY 200
[#IF NOT [#PROCESSEXISTS $SPLS]

| THEN | #OUTPUT *** ERROR *** PROCESS $SPLS WILL NOT START
| ELSE | #OUTPUT $SPLS Process Started...

]

] == Endif

```

Starting the Spooler

This subsection discusses the various methods of starting a spooler and provides example command files. The key distinction between a cold start and a warm start is that in a cold start the supervisor is run with a new control file, while in a warm start the supervisor is given the name of a control file that already exists. In operational terms, warmstarting the spooler recovers existing jobs and configurations and coldstarting the spooler erases any stored print jobs and previous spooler configuration settings.

-
- △ **Caution.** Always back up the Spooler control files before attempting to start the Spooler. This action preserves Spooler settings if there is a system problem and, if necessary, the backed-up control files can be available for analysis.
-

Warm Starting a Spooler

[Example 1-3](#) shows a command file that warm starts the spooler. Warmstarting a spooler recovers existing print jobs and configuration settings. Once the spooler has been brought up, the printer devices should be in the WAITING state. This command file can be invoked automatically from a system startup file (an example of which is in the *Himalaya S-Series Planning and Configuration Guide*), or you can invoke it by using the following TACL command:

```
> OBEY $SYSTEM.STARTUP.SPLWARM
```

Example 1-3. Spooler Warm Start File

```
comment -- This is $SYSTEM.STARTUP.SPLWARM
comment -- This file warm starts the spooler, leaving all jobs intact.
SPOOL / IN $SYSTEM.SPL.SPL, OUT $ZHOME, NAME $SPLS, NOWAIT, PRI 149, CPU 1&
/ 0
SPOOLCOM; SPOOLER, START
comment -- check to see that the spooler started successfully
SPOOLCOM; SPOOLER, STATUS
```

Note that \$ZHOME is specified as the home terminal in [Example 1-3](#). For G-series systems, \$ZHOME must be specified as the home terminal.

Cold Starting a Spooler

Coldstarting a spooler erases existing print jobs and configuration settings.

Before you cold start your spooler, you must purge all existing supervisor control files. (See [Cold Starting a Drained Spooler](#) on page 1-8.) Any jobs waiting in the current queue file are purged during a cold start, and users must respool these jobs after the cold start is finished.

Following are two command files that cold start a spooler in response to a single OBEY command. You can use these files as a model in setting up your own spooler cold-start command files. After you have entered the following files into two separate EDIT files, enter:

```
> OBEY $SYSTEM.SPLUTIL.COLDFILE
```

The commands contained in this file are executed. [Example 1-4](#) on page 1-6 shows the output of text that could appear on your home terminal:

Example 1-4. Spooler Cold Start File (COLDFILE)

```

COMMENT -- THIS IS $SYSTEM.SPLUTIL.COLDFILE
COMMENT -- THIS COMMAND FILE CREATES A SPOOLER SYSTEM
COMMENT -- Purge any existing collector data files having the
COMMENT same name as the data file you intend to use.
PURGE $MKT.SPL.DATAFILE
COMMENT -- Purge spooler supervisor control files having
COMMENT names you intend to use:
PURGE $MKT.SPL.SPL0
PURGE $MKT.SPL.SPL1
PURGE $MKT.SPL.SPL2
PURGE $MKT.SPL.SPL3
PURGE $MKT.SPL.SPL4
PURGE $MKT.SPL.SPL5
PURGE $MKT.SPL.SPL6
PURGE $MKT.SPL.SPL7
PURGE $MKT.SPL.SPL8
PURGE $MKT.SPL.SPL9
COMMENT -- Create a new collector data file:
FUP CREATE $MKT.SPL.DATAFILE, EXT ( 16384, 0 )
COMMENT -- Create the spooler supervisor process,
COMMENT specifying all cold-start parameters:
SPOOL / IN $MKT.SPL.SPL, OUT $ZHOME, NAME $SPLS, NOWAIT, &
PRI 147, CPU 0/1, 8191, 4095, 511, 10, 10, 40, 500
COMMENT -- supervisor control file = $MKT.SPL.SPL[0-9]
COMMENT -- log file for messages = $0 (op. console)
COMMENT -- supervisor process name = $SPLS
COMMENT -- execution priority = 147
COMMENT should be lower than editor or interactive utilities
COMMENT -- primary CPU = 0
COMMENT -- backup CPU = 1
COMMENT -- maximum number of jobs = 8191
COMMENT -- maximum number of locations = 4095
COMMENT -- maximum number of devices = 511
COMMENT -- maximum number of collectors = 10
COMMENT -- maximum number of print processes= 10
COMMENT -- maximum number of font jobs = 40
COMMENT -- maximum number of batch jobs = 500
COMMENT -- The spooler is now in the cold state; the supervisor is running.
COMMENT -- The next command runs SPOOLCOM, specifying a spooler
COMMENT "configuration file" to initialize the spooler:
SPOOLCOM / IN $SYSTEM.SPLUTIL.SPLCONF /
COMMENT -- Start the spooler:
SPOOLER, START

```

Note that \$ZHOME is specified as the home terminal in [Example 1-4](#).

The cold-start file in [Example 1-4](#) on page 1-6 causes SPOOLCOM to execute the commands contained in the following spooler configuration file named SPLCONF:

Example 1-5. Spooler Configuration File Named SPLCONF

```
COMMENT -- THIS IS $SYSTEM.SPLUTIL.SPLCONF

COMMENT -- THIS CONTROL FILE USES SPOOLCOM TO CONFIGURE THE
COMMENT SYSTEM TO BE COLD STARTED AND PASSES THIS
COMMENT INFORMATION TO THE SUPERVISOR.

COMMENT ***** YOUR SYSTEM
COMMENT ***** SPOOLER CONFIGURATION
COMMENT ***** 28 AUG 1993

COMMENT -- To configure one collector named "$S":

COLLECT $S, FILE $SYSTEM.SYSTEM.CSPOOL
COLLECT $S, DATA $MKT.SPL.DATAFILE
COLLECT $S, CPU 1, BACKUP 0, PRI 146

COMMENT -- To configure four print processes, one PSPool, one PSPoolB,
COMMENT one FASTP, and one
COMMENT user-written print process:

PRINT $STAND, FILE $SYSTEM.SYSTEM.PSPOOL, PRI 145
PRINT $STAND, CPU 0, BACKUP 1

PRINT $PSPB, FILE $SYSTEM.SYSTEM.PSPOOLB, PRI 145
PRINT $PSPB, CPU 0, BACKUP 1

PRINT $FSTP, FILE $SYSTEM.SYSTEM.FASTP, PRI 145
PRINT $FSTB, CPU 0, BACKUP 1

PRINT $USERP, FILE $USER.USER.USER, PRI 145, CPU 2

COMMENT -- To configure two line printers:

DEV $FAST, PROCESS $STAND, SPEED 900
DEV $SLOW, PROCESS $USERP, SPEED 300, WIDTH 120

COMMENT -- To configure one group, consisting of two locations:

LOC #DEFAULT.FAST, DEV $FAST
LOC #DEFAULT.SLOW, DEV $SLOW

EXIT
```

Cold Starting a Drained Spooler

This subsection explains how to cold start your spooler subsystem after it has been drained. See [Draining the Spooler](#) on page 1-9. Starting a new spooler is known as “cold starting” the spooler. Coldstarting a spooler erases existing print jobs and previously configured settings. (When you cold start a spooler, you create new control files in which the supervisor stores all the configuration information. When you warm start the spooler, the supervisor uses existing control files, and configuration settings are retained.)

Note. Cold starting a spooler subsystem is normally an infrequent task that you perform primarily in emergency situations. Before you perform a cold start, make certain that this is the appropriate operation.

The procedure for cold starting a spooler is generally the same from one time to the next. HP recommends that you maintain command files that can save time when you start the spooler.

To start a new spooler, log on as a super-group user (255, n)—not as the super ID (255,255)—and follow these steps:

1. You must either purge the datafile, or perform a FUP PURGEDATA on the datafile during a coldstart:

```
>PURGE data-file
```

or

```
>FUP PURGEDATA datafile
```

You must purge the file or issue PURGEDATA against the datafile. If you do not, the SPOOLCOM;COLLECT, STATUS DETAIL command, when issued after a coldstartup of spooler and after SPOOLCOM;COLLECT \$*collector-name*, START command, will show the percent full and allocated units of the datafile. If the datafile has not been newly created or has not had a PURGEDATA performed against it, the file will have an end-of-file (EOF) greater than zero.

2. Purge the supervisor control files.

```
> PURGE supervisor-control-filename
```

3. Create the data files in which collectors store jobs, one data file for each collector:

```
> FUP CREATE data-filename [ , create-param ]...
```

4. Run the spooler supervisor:

```
> SPOOL / IN control-filename , NAME $ supervisor-process /
```

5. Enter SPOOLCOM:

```
> SPOOLCOM
```

6. Specify the names and attributes of the collectors:

```
) COLLECT $ collector-name, DATA data-filename
```

7. Specify the names and attributes of the print processes:

```
) PRINT $ print-process
```

8. Specify the names and attributes of the devices:

```
) DEV $ device , PROCESS $ print-process
```

9. Specify locations and connect locations to devices:

```
) LOC [ # group ]. dest , DEV [ $device ]
```

10. Start the spooler:

```
) SPOOLER, START
```

Draining the Spooler

Draining brings the spooler to an orderly halt after all jobs have currently printed or spooling has finished. When the spooler is drained, the Supervisor updates its control file, halts, and enters the dormant state, ready to be warmstarted. If you improperly drained the spooler, see [Recovery for an Improperly Drained Spooler](#) on page 1-12.

In summary, these are the steps needed to properly drain the spooler.

Step 1: Check for Jobs That Are Still in the Open or Print State

1. Log on as a super group user (255,n).
2. Enter SPOOLCOM at the TACL prompt, as follows:

```
TACL>SPOOLCOM
```

3. Enter the COLLECT command with the LISTOPENS qualifier, as follows:

```
SPOOLCOM>COLLECT $col-process,LISTOPENS
```

Jobs that are currently printing should be allowed to complete printing or be placed in a HOLD state. The spooler supervisor cannot complete the drain task until all the jobs are closed and have completed printing.

To hold a job, issue the following command:

```
SPOOLCOM>JOB job-number, HOLD
```

To hold all jobs currently printing, issue this command:

```
SPOOLCOM>JOB (STATE PRINT),HOLD
```

Step 2: Drain the Spooler Properly

To drain the spooler, log on as a super group user (255,n), access SPOOLCOM, and enter the following command:

```
SPOOLCOM> SPOOLER, DRAIN
```

This command causes the spooler to move from the active to the dormant state.

Caution. Do not use the TACL STOP command to bring down the spooler. Also, never use the TACL STOP command after you have issued the SPOOLCOM DRAIN command. The warmstartup recovery or rebuild from anything less than a successful drain can be both time consuming and the rebuild can corrupt the control file. The risk of corrupting the control file is even more certain if you are migrating from older spooler versions.

Note. During this time when both the Supervisor and spooler processes are dormant, you *cannot* obtain any information regarding the configuration of the spooler, that is, the collectors, the number of devices and print processes that are attached to the Supervisor.

Step 3: Back up the Spooler Control and Data Files

While the spooler environment is closed, you should back up the spooler control and data files. Backing up the spooler will render a known database from which you can recover in case of a catastrophic spooler failure.

△ **Caution.** Do not back up spooler control and data files *while* they are in the open state. If you do, the control file could either become corrupt or no longer synchronized with the data file; and using these files for subsequent recovery and warmstart could lead to serious spooler problems.

Steps for Backing Up the Control and Data Files

The following steps describe how to identify and locate the spooler control and collector data files so that they can be properly backed up.

1. Identify the spooler control and collector data files. If you do not know the location of the spooler control and collector data files, you can use the following commands to locate them. If you have multiple spoolers, you will need to follow these procedures for each spooler.
2. If you do not know the location of the spooler control files (128), perform the following FUP INFO command:

Example 1-6. FUP INFO Command To Find Spooler Control Files

```
34> FUP INFO $*.*.* WHERE FILECODE = 128
          CODE EOF          LAST MODIF          OWNER    RWE P    TYPE    REC    BL
$SYSTEM.SPOOLER
CTRL3  O          128  4036608    20Oct1999 23:36    -1      NNNN
```

You also must know the name of the Supervisor process that has the control file open. To identify the Supervisor process, you need to:

- Find the PID of the Supervisor process. Issue the following FUP command:

```
35> FUP LISTOPENS $SYSTEM.SPOOLER.CTRL3
```

[Example 1-7](#) shows the outcome of this command:

Example 1-7. PID of the Supervisor Process

```
$SYSTEM.SPOOLER.CTRL3
```

PID	MODE	USERID	SD	MYTERM	PROGRAM FILE NAME
248,01,0078-P	R/W-P	255,255	01	\$ZHOME	\$SYSTEM.SYSTEM.SPOOL
248,00,0093-B	R/W-P	255,255	01	\$ZHOME	\$SYSTEM.SYSTEM.SPOOL

Using the PID, find the process name, as shown in [Example 1-8](#).

Example 1-8. Using the STATUS Command

```
36>STATUS,1,78
```

```
System \BBQ
```

Process	Pri	PFR	%WT	Userid	Program file	Hometerm
\$SPL1	1,78	160	001	255,255	\$SYSTEM.SYSTEM.SPOOL	\$ZHOME
Swap File Name: \$SYSTEM.#0						
\$SPL1 B	0,93	160	001	255,255	\$SYSTEM.SYSTEM.SPOOL	\$ZHOME
Swap File Name: \$SYSTEM.#0						

Knowing the supervisor process name allows you to find the collector data files as shown in [Example 1-9](#):

Example 1-9. Finding the Collector Files

```
37>SPOOLCOM $SPL1;COLLECT
```

COLLECT	STATE	FLAGS	CPU	PRI	UNIT	DATAFILE	%FULL
\$S	ACTIVE		1	155	4	\$SYSTEM.SPOOLER.DATAS	63
\$C	ACTIVE		1	155	8	\$DATA01.SPOOLER.DATAC	28

- You can now drain the spooler. Refer to [Draining the Spooler](#) on page 1-9.
- Back up the files identified in step 1. The following command will back up all files in subvolume \$SYSTEM.SPOOLER, including CTRL3 and DATAS. This command also backs up \$DATA01.SPOOLER.DATAC files:

```
BACKUP/OUTSPLBKUP/$TAPE, ($SYSTEM.SPOOLER.*, $DATA01.SPOOLER.DATA), LISTALL
```

Recovery for the Spooler

Use this subsection to help recover from a spooler problem.

Recovery for an Improperly Drained Spooler

If a spooler was improperly drained or suffered an abnormal shutdown, follow these steps to recover from this situation:

1. Warm start the spooler.
2. Enter the SPOOLCOM JOB command:


```
) JOB
```
3. If you find a Job 0 listed as a result of the SPOOLCOM JOB command, the spooler has a corrupted job map.

Perform the steps described in [Rebuild the Spooler Control Files and the Job Bit Map](#).

Rebuild the Spooler Control Files and the Job Bit Map

To rebuild the spooler control files and the job bit map, execute the same SPOOL command you normally use for a warm start but add the REBUILD option after the supervisor process name, and specify new values. These values should be the same as those specified in your cold-start command file. To rebuild the spooler control files, follow these steps:

1. Drain the spooler as described in [Draining the Spooler](#) on page 1-9:

```
> SPOOLCOM SPOOLER, DRAIN
```

2. Rebuild the spooler control files:

```
> SPOOL / IN control-filename ,
      NAME $supervisor-process-name /
      R [EBUILD]

      , backup-cpu
      , num-of-jobs , num-of-locations
      , num-of-devices , num-of-collectors
      , num-of-print-processes
      , num-of-font-jobs
      , num-of-batch-jobs
```

You must include the REBUILD option if you want to increase the values of any cold-start parameters (*num-of-jobs* and so forth) in a warm start. If you do not use this option, you must cold start the spooler to increase the values of any parameters. You cannot decrease the values of any parameters.

3. Optionally add, delete, or modify spooler components.

4. Enter SPOOLCOM and start the spooler:

```
> SPOOLCOM SPOOLER, START
```

This example shows how to rebuild control files.

```
> SPOOLCOM SPOOLER, DRAIN  
> SPOOL /IN $MKT.SPL.SPL, NAME $SPLS/ R,995,995,150,10,10  
> SPOOLCOM SPOOLER, START
```

The parameters used in this example are as follows:

- \$MKT.SPL.SPL is the control file.
- \$SPLS is the name of the supervisor.
- R stands for REBUILD.
- 995 is the maximum number of jobs and locations for this spooler.
- 150 is the maximum number of devices for this spooler.
- 10 is the number of collectors and print processes for this spooler.

Connecting Spoolers in a Network Using RPSetup

The RPSetup utility connects spoolers in a network to give access to printers on remote nodes. You use RPSetup to create and maintain a spooler configuration for a node with a limited number of printers so that print jobs from local applications can be sent to a remote spooler and queued for printing on remote printers. RPSetup reads configuration information from the remote spooler supervisor, analyzes the information, and writes new configuration information to the local spooler supervisor.

RPSetup sets up connections to all valid locations on the remote node. To create these connections, RPSetup creates locations on the local spooler with the same names as locations on the remote node. RPSetup might also add required print processes and devices to the local spooler configuration.

Enter the following command to run RPSETUP:

```
> RPSETUP / IN remote-spooler , OUT local-spooler /
```

For detailed information on RPSetup, see [Section 6, RPSetup Utility](#).

2

Introduction to the Spooler Subsystem

The Spooler subsystem is a set of utilities that form an interface between you (and your application programs) and the print devices of a system. The Spooler subsystem receives output from an application and stores it on disk. This output can be a report in the format of an EDIT file (file code 101), a listing from a compiled program, or other data. When the designated print device becomes available, the output is printed.

Topics described in this section include the following:

- [Spooler and Spooler Plus Comparison](#) on page 2-2
- [Spooler Features](#) on page 2-2
- [Why Use the Spooler?](#) on page 2-2
- [Spooler Components](#) on page 2-4
- [Spooling in a Network](#) on page 2-8
- [Peruse and Spoolcom Comparison](#) on page 2-10
- [Spooler Jobs and Job Attributes](#) on page 2-10
- [Devices and Device Attributes](#) on page 2-13
- [Routing Structure](#) on page 2-16
- [Using the Spooler](#) on page 2-18
- [The Spooler and Batch Jobs](#) on page 2-19
- [Spooler Security](#) on page 2-23

Spooler and Spooler Plus Comparison

Spooler Plus is an optional product containing Spoolcom and Peruse modules that can be used to replace the Spoolcom and Peruse modules provided by the D41 or later releases of the Spooler subsystem. You can use the Spooler Plus Spoolcom and Peruse utilities to configure and manage expanded configurations of the Spooler subsystem. You *must* use these utilities if the maximum jobs has been configured above 8191.

The Spooler Plus subsystem is described in the *Spooler Plus Utilities Reference Manual* and the *Spooler Plus Programmer's Guide*.

Spooler Features

Features of the Spooler subsystem include

- Continuous operation—the Spooler subsystem keeps working even if a processor fails.
- Flexible routing—you can change the destination of a job even after it enters the Spooler subsystem.
- No programming necessary—you can send output to the Spooler subsystem simply by specifying a spooler location as your OUT file.
- Interactive control—using the Peruse program, you can inspect or alter the status of your job, examine the data you have sent to the Spooler subsystem, and change the destination of your output.
- Operator control—using the Spoolcom utility, you can initialize and modify all Spooler subsystem components.

Why Use the Spooler?

Note. Because **spooler** is an industry-standard term used to describe a printer spooling system, it used throughout the remainder of this manual to refer to the Spooler subsystem.

The spooler offers a consistent interface to all print devices in your system. The application programs (such as TAL, TFORM, or COBOL85) can send data to the spooler as easily as they do to a disk file. The spooler ensures efficient use of the print devices in the system because it allows many different programs to send data to the same printer at the same time while protecting the applications from device-dependent considerations.

There are alternatives to using the Stop spooler, and in some special circumstances these can be more efficient. For most users, however, using the spooler is the best way to produce hard-copy output.

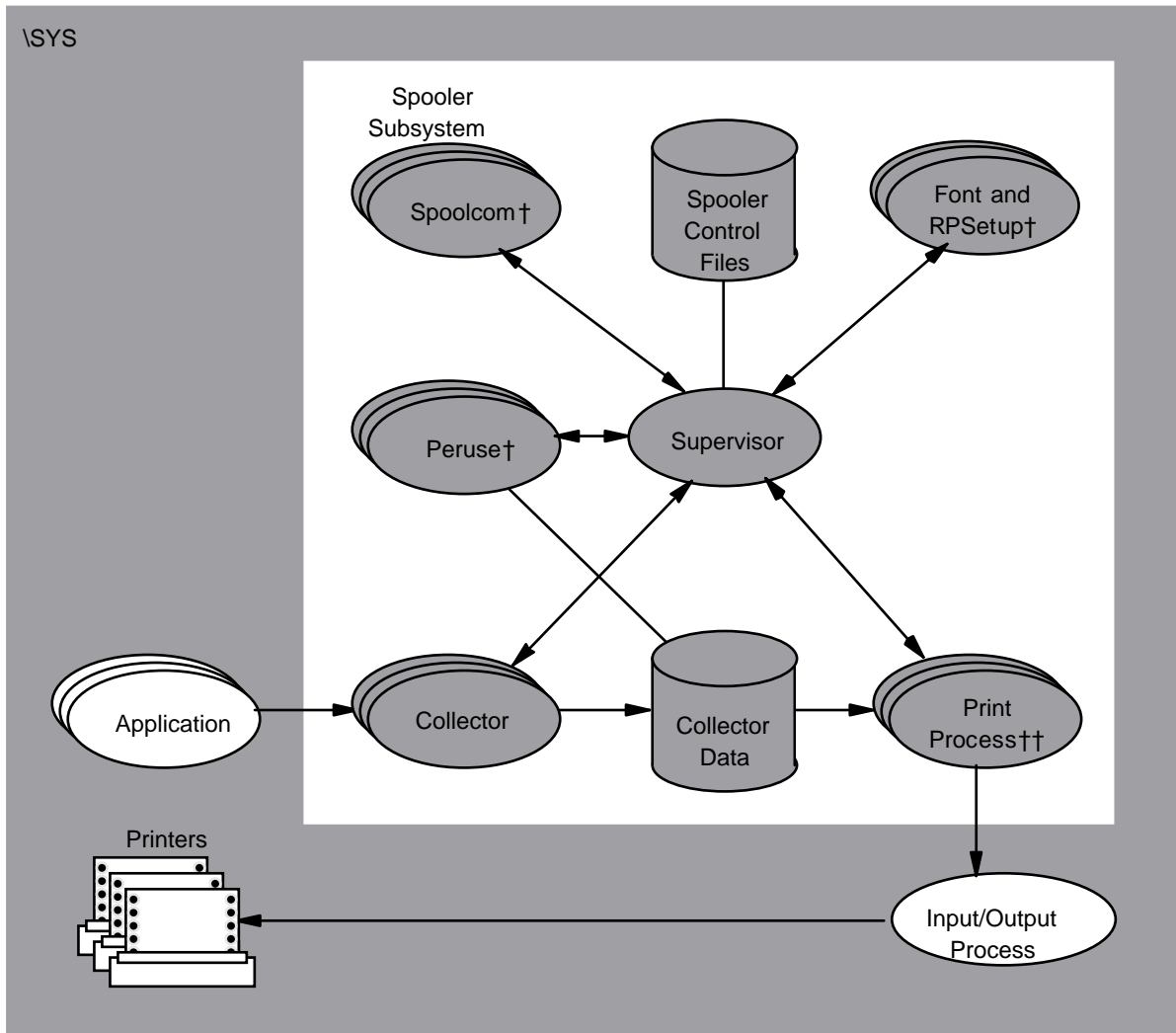
Spooler Components

The spooler consists of the components:

- The spooler supervisor monitors and communicates with the other programs and determines when and where to print jobs. There is only one supervisor in each spooler subsystem; however, many spooler subsystems can operate at one time.
- Collector processes accept output from applications and store it on disk. There can be up to 15 collectors associated with a given spooler. Collector processes are described in [Collectors](#) on page 2-6.
- Print processes retrieve the output stored on disk by a collector and print it on a device. Each print device in the spooler subsystem has a print process associated with it. FASTP is the print process released with the spooler. For more information on FASTP, see [Appendix F, Using FASTP](#). You can also write your own print process. See the *Spooler Programmer's Guide* for details.
- Peruse allows you to control and monitor your jobs. You run Peruse interactively from a terminal. Peruse translates your commands into messages to the supervisor, which then carries out your instructions. Complete Peruse command syntax is shown in [Section 3, Peruse](#).
- Spoolcom is the interface to the spooler subsystem. You can use Spoolcom interactively to determine the status of all spooler components, to start a device that is offline, and to send instructions to a device (to skip pages between printing jobs, for example). System operators use Spoolcom to create and initialize the components of the spooler subsystem. Complete Spoolcom command syntax is shown in [Section 4, Spoolcom](#).
- The Font utility assists in creating font jobs. To configure spooler locations, you can enter font job descriptions in an EDIT file, called a script file, and run the Font utility, which configures those spooler locations with the indicated font jobs. A complete description of the syntax for the script file and the command to run Font are in [Section 5, Font Utility](#).
- The RPSetup utility assists in maintaining spooler configurations in a network. A complete description of RPSetup configuration parameters and the command to run RPSetup are in [Section 6, RPSetup Utility](#).

The spooler components are shown in [Figure 2-1](#)

Figure 2-1. Components of the Spooler



Legend

- † Applications can perform all the functions of Peruse, Spoolcom, Font, and RPSetup; therefore, these programs can be considered applications.
- †† A print process can be FASTP or user written.

CDT 019CDD

Collectors

The spooler includes one or more collectors, each of which is a continuously running copy of the program in `$SYSTEM.SYSTEM.CSPOOL`. An application can direct output from an application program to a collector by treating the collector as its OUT file; that is, an application can open a file to any collector and begin writing its output using the Guardian file-system `WRITE[X]` procedure. Applications can also use the spooler interface procedures to spool their output (described in the *Spooler Programmer's Guide*).

While the spooler is in the COLD or WARM state, you can declare and initialize the spooler collectors by using the Spoolcom `COLLECT` command, specifying attributes such as execution priority and program file name.

Collector Attributes

[Table 2-1](#) lists the default attributes of collectors and the Spoolcom `COLLECT` subcommands used to specify them.

Table 2-1. Collector Attributes

Collector Attribute	Spoolcom <code>COLLECT</code> Subcommand	Default Value
Program file	FILE	<code>\$SYSTEM.SYSTEM.CSPOOL</code>
Primary processor	CPU	Processor of supervisor
Backup processor	BACKUP	No backup processor
Execution priority	PRI	145
Data file	DATA	None
Unit size	UNIT	4
Page size	PAGESIZE	60

Optimizing Collector Performance

The collector file size must be less than 2 GB including all partitions. The file's buffer size is used to set the collector's internal buffer size. The buffer size must be at least 1K bytes and the unit size must be at least 2K bytes. The unit size must be the same as or a multiple of the buffer size, or the buffer size must be the same as or a multiple of the unit size. The recommended configuration is a buffer size of 4K bytes and a unit size of 4K bytes.

The collector allows the usage of all available file space. It uses an index block to account for the space used within a job. Each index contains 220 entries and multiple chained index blocks are used when needed. Concurrent users are allocated space at different points in the collection file. This tends to cause allocation within a job to occur in continuous groups of units. Because each index entry can represent from 1 to 512 contiguous units, each index block theoretically could represent up to 112,640 units.

Index blocks are used to keep track of the units of space allocated to a job. The current index block for a job is kept in memory during collection so that there is no overhead as entries are being made. The collector chooses the addresses of the first allocated unit in the collection file for a job in such a way as to minimize the contention for the same units by different jobs. It also attempts to keep the units for a job as contiguous as possible.

The first index block for a job can contain up to 198 entries. Subsequent index blocks for the job can contain up to 207 entries. When an index block fills, another is established and two additional disk input-output (I/O) operations will result. Each entry in an index block can depict from 1 to 512 units of contiguous space. For example, if a job has 500 units of space allocated to it and all that space is contiguous, only a single entry in a single index block is required to account for the job's use of space. This ability to combine multiple allocations into a single entry minimizes the number of index entries required and therefore minimizes the number of index blocks required for a job.

When PRINTREAD is requested to start reading at a specific page within a job (for example, from the Peruse JOB LIST *page-num* and Spoolcom DEV SKIPTO *page-num* commands), the range of pages represented by an entire index block is used to determine if another index block must be read. The number of additional I/O operations required for the start of the operation will be, on average, half of the number of additional index blocks for the job.

The worst-case performance impact of index blocks is no more than .5 percent during collection. This can occur where the unit size is 1K bytes and every allocated unit is not contiguous to the previous unit. For jobs containing 500 units or less, there should be no noticeable impact on performance.

For random page access, the worst case performance impact is an additional I/O per index block before the read can begin. Reading a job from the beginning, or reading the job after the selected page has been found, does not involve index blocks.

The large number of entries per index, the tendency toward allocating contiguous units, and the combining of contiguous allocations combine to greatly reduce the impact of index blocks on performance because there will generally be no need for extra index blocks.

As a general rule, unit sizes should be small to enable more efficient use of collector space. Unless the typical job is much less than 4K bytes, a unit size of 4K is recommended along with a 4K buffer size. The impact of too small a unit size (jobs with 50 or more index blocks) is not noticeable during collection and is nonexistent during delivery. Too-small unit size should be considered as a potential performance problem only when performance of Peruse JOB LIST or Spoolcom DEV SKIPTO performance is especially important.

The collector monitors space usage when units are allocated and deallocated. If a change of .5 percent or greater has occurred since the last usage report, a special report is sent to the supervisor.

Each collector process handles as many as 1024 simultaneous users. Consider employing a smaller number of collector processes.

Unit Size

The Spoolcom COLLECT UNIT command is used to specify a unit size for the collector. The unit size specifies the number of 512-word blocks the collector allocates from its data file each time it needs more space for a job. A collector can manage as many units as a physical file will hold.

The larger the unit size, the less often the collector must allocate a new unit. Thus, you could specify a relatively large unit size if you expect that most spooled jobs will be large. You could also use a large extent size if using large block sizes.

A smaller unit size provides more efficient use of disk space, because once the collector reserves space for a job, that space cannot be used by any other job. If the unit size is 10 and a spooled job requires only 1 block, the other 9 blocks are wasted.

You should set the unit size of a collector once and not change it. If a different unit size is required, delete the old collector and start a new one.

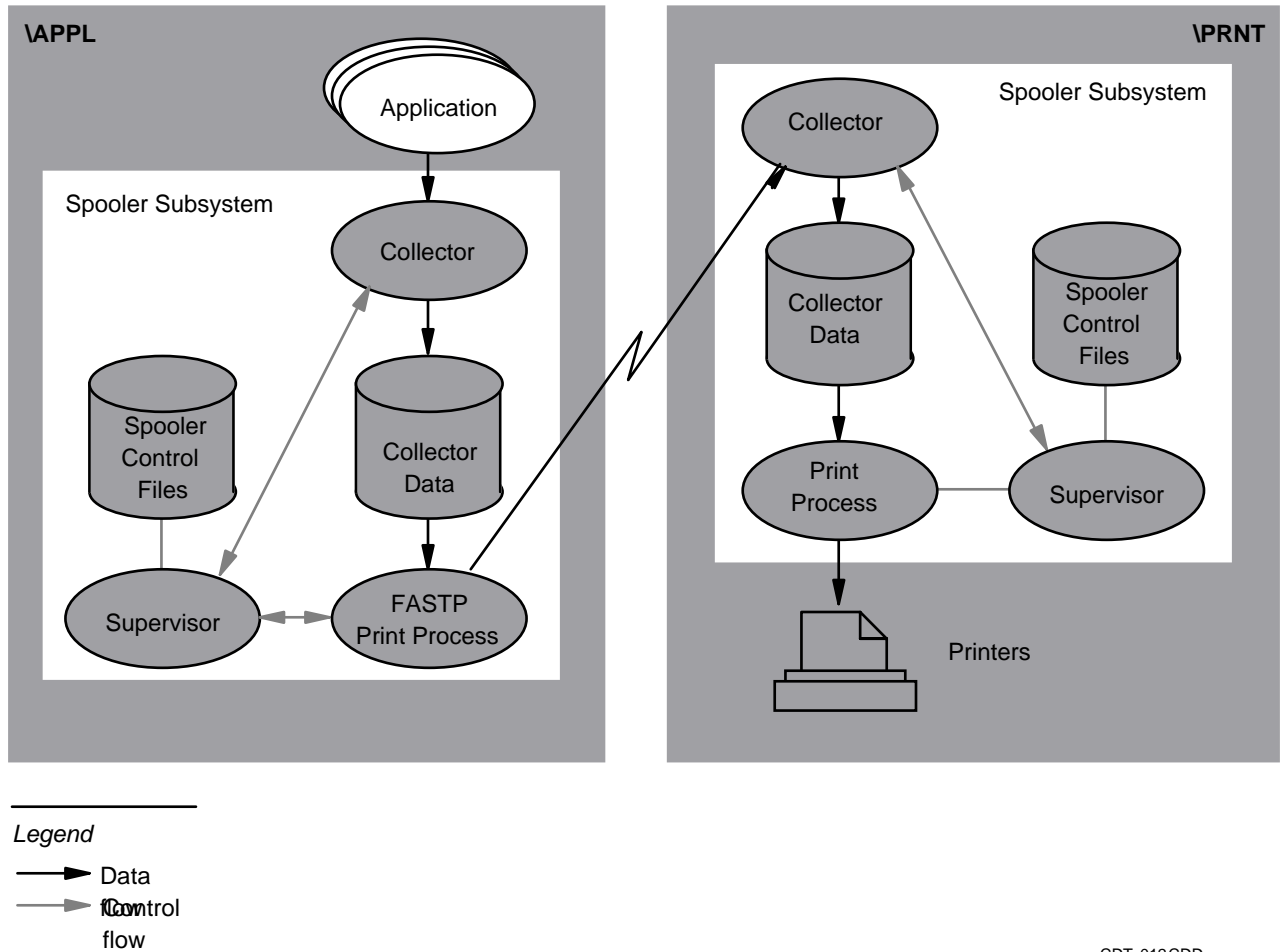
The ability to specify different unit sizes with different collectors is consistent with pre-D41 spoolers. However, for the D41 and later releases of the spooler, HP recommends a unit size of 4K bytes and a DP2 buffer size of 4K bytes, unless jobs are likely to be less than 2K bytes in length. The spooler is most efficient when 4K values are used, but if jobs are generally less than 2K bytes, a 2K setting for both unit size and buffer size will yield about the same performance and be more efficient in the use of disk space. Note that 2K bytes is the minimum value you can specify for buffer size.

Spooling in a Network

Spoolers can be linked in a network such that applications on one node can have their jobs printed on another node with the spooler managing the print-job transfers across the network. The link between network spoolers is illustrated in [Figure 2-2](#). On the application node \APPL, applications send print jobs to their local spooler. The spooler is configured to have the FASTP print process control the transferring of print jobs to the remote node \PRNT, where the actual printers are located. The print process controlling the printers could be any spooler-supplied or user-written print process.

For information on managing networks linked as illustrated in [Figure 2-1](#), see [Section 6, RPSetup Utility](#).

Figure 2-2. Linked Spoolers in a Network



CDT 012CDD

Peruse and Spoolcom Comparison

Many commands in Peruse and Spoolcom are nearly equivalent. For example, the Peruse DEV and Spoolcom DEV commands both display the status of devices, and many Peruse commands are also subcommands of the Spoolcom JOB command (COPIES, DEL, FORM, HOLDAFTER, OWNER, and PRI). However, Peruse and Spoolcom have significant differences:

- Peruse provides a convenient means for monitoring and changing your jobs. When you enter Peruse, you immediately see a list of the jobs you own in the spooler subsystem, identified by job number and other job attributes. You can examine a spooler job page-by-page or line-by-line, and you can use the FIND command to locate specific character strings. With Peruse, you can use fewer keystrokes than with Spoolcom to change a job attribute, send a job to a print device, or delete a job.
- Spoolcom is a tool that system operators use for monitoring and controlling the spooler subsystem. All users can use Spoolcom to obtain status information on the spooler components and on all spooler jobs in the system. In addition, all users can use Spoolcom to change attributes of their own jobs, but this is done more conveniently with Peruse.

Spooler Jobs and Job Attributes

When you request the spooler to print information, the request is called a spooler job. The spooler assigns to each job a job number in the range 1 through 4095. (The system operator can set the maximum job number to a lower value with the SPOOL program. See [Appendix A, SPOOL Program](#).) In addition to the job number, jobs have seven primary attributes:

- Batch number
- Job priority
- Job copies
- Job report name
- Job form name
- Job location
- Job state

Peruse allows you to display and alter these attributes. Descriptions of these attributes follow.

Batch Number

The batch number identifies a batch job, which is a group of individual jobs that have been linked together by the spooler or by the Peruse LINK command. Batch jobs are described in detail in [The Spooler and Batch Jobs](#) on page 2-19.

Job Priority

Job priority determines when a job will print in relation to other jobs queued for the same device. The spooler maintains a device queue for all print devices, and higher-priority jobs are placed ahead of lower-priority jobs in the queue. For more information, see [Selection Algorithm](#) on page 2-15.

Job Copies

The job copies attribute specifies the number of copies to print.

Job Report Name

The job report name is printed in banner-sized letters in the header message of the job. The header message is described later in this section under [Devices and Device Attributes](#) on page 2-13.

Job Form Name

The job form name is an optional attribute that allows you to ensure that your job is printed only on a device that has been loaded with the correct paper or forms. A job with a job form name will print only on a device that has a corresponding device form name. (Refer to [Devices and Device Attributes](#) on page 2-13.)

For example, if an application program produces a job that fills out W-2 forms, the job must be printed on a device loaded with special W-2 form paper. If the job has a form name of W2, it will print only on a device that has the same form name. The form name W2 is assigned to the device when the special paper is loaded.

Job Location

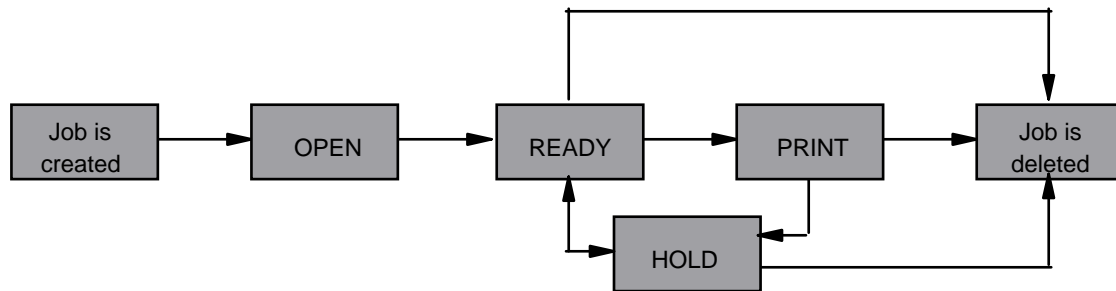
The job location can be associated with a physical print device on the system or with a dummy holding location. Job location is described in detail later in this section under [Routing Structure](#).

Job State

Job status is described by the job state. A job can be in one of four states:

- OPEN
- READY
- PRINT
- HOLD

[Figure 2-3](#) illustrates the job states and how they relate to the life cycle of a spooler job.

Figure 2-3. Life Cycle of a Spooler Job

CDT 003CDD

You create a job when your application opens a collector process and writes to it; for example:

```
TFORM / IN filename, OUT $S.device /
```

A job in the OPEN state is being added to the spooler. For example, if you send the output from a TFORM process to the spooler, the job is in the OPEN state until the TFORM output is complete.

A job in the READY state is ready to print but is not yet printing. The job enters the READY state when the application closes the collector process.

The job waits in the device queue until it is ready to print. A job in the PRINT state is being printed. The PRINT state is normally the last state before the job is deleted from the spooler. To prevent the spooler from deleting a job after printing, use Peruse to set the hold-after-printing flag; see [Section 3, Peruse](#), for details.

You can put a job in the HOLD state by entering the Peruse HOLD command. A job in the HOLD state is not printed and remains in the spooler indefinitely until you delete it or remove the HOLD state.

You can put a job in the HOLD state at any time. If the job is in the READY or PRINT state, it can be placed in the HOLD state immediately. If it is in the OPEN state, it is placed in the HOLD state after the application output is complete.

The life cycle of a print job starts with the OPEN state, which continues while the application writes the data to the collector and the collector stores the data in a disk file. When the application has sent all the data to the collector, the job is in the READY state. The job then enters the PRINT state if its spooler location is associated with a device. (If the location is not associated with a device, the job enters the HOLD state.) When printing is complete, the spooler deletes the job unless the hold-after-printing flag is set. You can also delete a job from the spooler by using a specific request. If the job is printing when you make that request, the job immediately stops printing and is deleted.

Devices and Device Attributes

A device produces a hard-copy listing of your job. Every device is controlled by a print process. Devices (and print processes) have four attributes that can affect your spooler jobs:

- Form name
- Device header message
- Device state
- Selection algorithm

Device Form Name

The form name of a device is an optional attribute that allows you to control the type of job that can be printed on the device. The device form name indicates the type of paper or forms that are loaded in the device. Only a job with a job form name that corresponds to the device form name can be printed on the device.

Device Header Message

The device header message includes the job report name, location, job number, form name, and date and time of printing. You can turn the header message on or off for each device. When the header message is turned on, it prints on the first page of the job; the report name and location are printed in large banner-sized letters, as shown in [Figure 2-4](#). If the header message is turned off, jobs print consecutively with only a form feed (new page) to indicate the beginning of the next job.

Figure 2-4. Sample Header Page Produced by the FASTP Print Process

```

AAA          DDDDDDDDDDD      MMM      MMM      III      NNN      NNN
AAA AAA      DDDDDDDDDDD      MMMM     MMMM     III      NNNN     NNN
AAA  AAA     DDD      DDD      MMMMM  MMMMMMM  III      NNNNN   NNN
AAA AAAA     DDD      DDD      MMMMMMMMMMMMM  III      NNNNNN   NNN
AAAAAAAAA    DDD      DDD      MMM  MMM  MMM      III      NNN  NNN  NNN
AAAAAAAAA    DDD      DDD      MMM  M  MMM      III      NNN  NNNNNN
AAA  AAA     DDD      DDD      MMM      MMM      III      NNN      NNNNN
AAA  AAA     DDDDDDDDDDD      MMM      MMM      III      NNN      NNNN
AAA  AAA     DDDDDDDDDDD      MMM      MMM      III      NNN      NNN
AAA  AAA     DDDDDDDDDDD      MMM      MMM      III      NNN      NNN

```

```

LLL          EEEEEEEEEEE      EEEEEEEEEEE
LLL          EEEEEEEEEEE      EEEEEEEEEEE
LLL          EEE              EEE
LLL          EEE              EEE
LLL          EEEEEEE         EEEEEEE
LLL          EEEEEEE         EEEEEEE
LLL          EEE              EEE
LLL          EEE              EEE
LLLLLLLLLLLL EEEEEEEEEEE      EEEEEEEEEEE
LLLLLLLLLLLL EEEEEEEEEEE      EEEEEEEEEEE

```

```

###  ###      HHH      HHH      TTTTTTTTTTT      333333333
###  ###      HHH      HHH      TTTTTTTTTTT      33333333333
#####      HHH      HHH      TTT              333      333
#####      HHH      HHH      TTT              333      333
###  ###      HHHHHHHHHHH      TTT              3333
###  ###      HHHHHHHHHHH      TTT              3333
#####      HHH      HHH      TTT              333      333
#####      HHH      HHH      TTT              333      333
###  ###      HHH      HHH      TTT              33333333333
###  ###      HHH      HHH      TTT              333333333

```

```

HHH      HHH      TTTTTTTTTTT      33333333333
HHH      HHH      TTTTTTTTTTT      33333333333
HHH      HHH      TTT              333      333
HHH      HHH      TTT              333      333
HHHHHHHHHH      TTT              3333
HHHHHHHHHH      TTT              3333
HHH      HHH      TTT              333      333
HHH      HHH      TTT              333      333
HHH      HHH      TTT              33333333333
HHH      HHH      TTT              333333333

```

Date: 03 June 92, 14:01:07

JOB: 1919

FORM:

If you specify a batch header, the job information prints on two of the three trailer pages as well as on the first two pages of each job. The trailer pages have printing over the page folds, enabling jobs printed on accordion-fold paper to be separated easily. The two-page header message always appears on the top page, regardless of how the job is folded.

The actual header message produced depends on the print process controlling the device. The headers described above are produced by the Tandem print process. If a device is controlled by a user-written print process, the process can produce almost any kind of header or none at all.

Device State

The device state describes the status of the device. There are six possible device states:

PRINTING	The device is currently printing a job.
WAITING	The device is idle and is waiting for a job to print.
OFFLINE	The device is not available for printing.
SUSPENDED	The device is in the process of printing a job but has been suspended by the operator (to change ribbons, for example).
DEVEERROR	The device has produced a file-system error while printing. Operator intervention is required.
PROCERROR	The supervisor process has determined that the print process for the device is not working correctly. Operator intervention is required.

Selection Algorithm

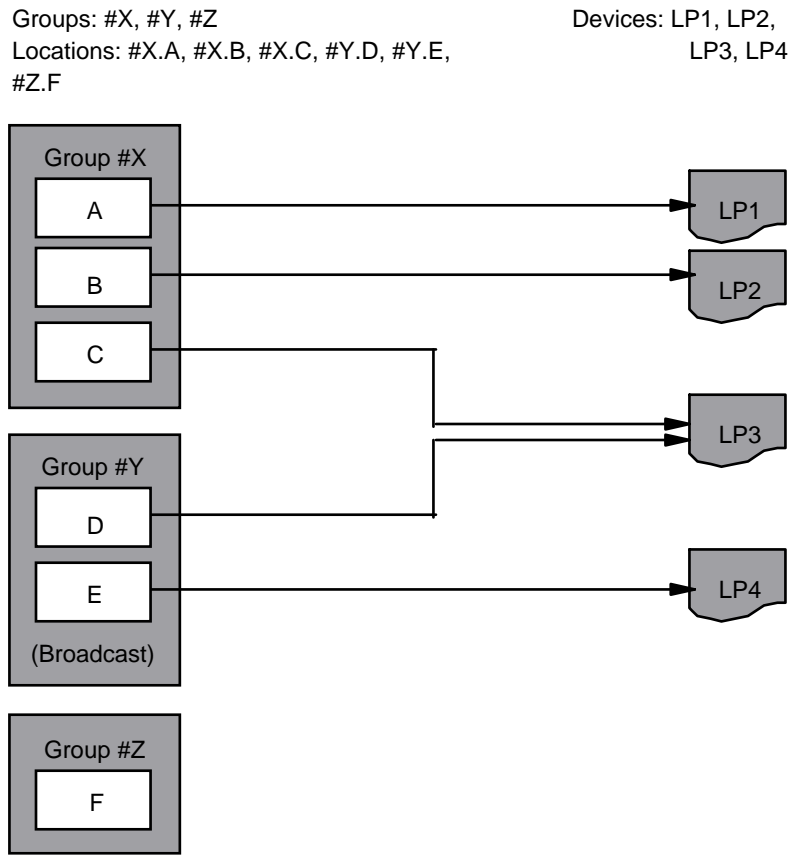
The spooler maintains a queue for each device (that is, a list of the jobs to be printed on that device.) The job at the head of the queue is the next job to be printed.

As a rule, higher-priority jobs print sooner than lower-priority jobs. However, the selection algorithm affects the order in which jobs print within the same priority level. If the selection algorithm is FIFO (first in, first out) ON, jobs are placed at the end of the queue and are printed in order. If the selection algorithm is FIFO OFF, the spooler allows short jobs to print before longer jobs of the same priority.

Routing Structure

The routing structure directs jobs to print devices. It consists of a set of locations and print devices. [Figure 2-5](#) shows the association of locations with print devices.

Figure 2-5. The Spooler Routing Structure



CDT 003CDD

A spooler **location** is the logical destination of a job; a **print device** is the physical destination. This distinction permits great flexibility when routing jobs. At the time a job enters the spooler subsystem, the spooler assigns it to the location that was designated by the OUT option in the RUN command that started the application. (Refer to the description of the RUN command in the *TACL Reference Manual*.) If there is a device associated with the assigned location, the job eventually prints on that device.

Location names have two parts: a **group name** and a **destination name**. The group name is always preceded by the # character; for example, #LP and #HT. Examples of full location names are:

```
#LP.LPEAST  
#LP.LPWEST
```

#LP is a group name, and LPWEST and LPEAST are destination names.

Broadcast and Nonbroadcast Groups

If you specify only the group name as the location, the spooler supplies the destination. If the group is a nonbroadcast group, then the spooler routes the job to the destination that can print the job soonest. If the group is a broadcast group, then the job is routed to all the destinations in the group and prints on all the devices associated with the group.

For example, assume that the location name #LP.LPEAST is associated with a line printer on the east side of the machine room, and #LP.LPWEST is associated with a line printer on the west side of the machine room. If #LP is a broadcast group, then a job routed to #LP prints on both line printers, and two copies of the job are printed. If #LP is a nonbroadcast group, then the first available line printer prints the job and only one copy of the job is printed. In either case, a job routed to #LP.LPWEST prints only once on the line printer at the west end of the machine room.

Your system operator can tell you which locations are available to you, which print devices are associated with those locations, and which groups are broadcast groups.

A printer can be connected to several locations, but each location is connected to no more than one printer.

Default Routing

The spooler has a special location, #DEFAULT, which is used when you do not specify a location for a job. For example, these two commands are equivalent; both send TFORM output to \$S.#DEFAULT:

```
1> TFORM / IN DAYREPT, OUT $S/  
1> TFORM / IN DAYREPT, OUT $S.#DEFAULT /
```

Consult your system manager to determine which physical device or devices are associated with #DEFAULT on your system.

Implicit Route Creation

When jobs are routed to a dummy location, the spooler implicitly creates a route. (A **dummy location** is a nonexistent location.) For example:

- When a job is routed to location #X.1, where either the group or the destination does not exist, the spooler creates the location #X.1.
- When a job is routed to group #X, where group #X does not exist, the location #X.DEFAULT and the group #X are created.

Many users choose the location group #HOLD as a holding location so that they can use Peruse to examine their jobs before printing. You can use #HOLD or any dummy location for this purpose. However, to print the job, you must use the Peruse LOC command to change the location. Alternatively, you can use the Spoolcom JOB command.

Using the Spooler

To use the spooler from the Tandem Advanced Command Language (TACL), you send process output either to a spooler location or to a SPOOL DEFINE. In either case, you normally specify the spooler destination as the OUT file.

Sending Process Output to a Spooler Location

You can send a job directly to the spooler by designating a collector (such as \$S) and a location (such as #LP.LPWEST) as the OUT file when you run a program. For example, the following command spools the output from a TFORM process:

```
1> TFORM /IN MYFILE, OUT $S.#LP.LPWEST/
```

The collector \$S creates the job, assigns a job number, and stores the TFORM output in a disk file. When TFORM has finished sending data, the spooler puts the job in the ready state. If there is a print device associated with #LP.LPWEST, the job is printed when that device is free. If there is no print device at that location, the job remains in the spooler until you delete it or send it to an output device.

You do not always need to specify the entire location name. The following command is sufficient:

```
2> TFORM /IN MYFILE, OUT $S.#LP/
```

If #LP is a nonbroadcast group, the job is printed on the first available print device associated with the #LP group.

For location groups that have only one associated print device, the full location name is also unnecessary. The following example shows a TFORM command that does not specify the location:

```
3> TFORM /IN MYFILE, OUT $S/
```

In this case, the job is sent to #DEFAULT and is printed on one of the devices associated with the #DEFAULT group name.

Sending Process Output to a SPOOL DEFINE

You can also use a SPOOL DEFINE to specify the attributes of a spooler job before you create it. Specifying all the spooler job attributes in a SPOOL DEFINE is faster than using Peruse to change the job attributes after the job is in the spooler.

You specify the name of the SPOOL DEFINE as the OUT file when you run your program. For example, the following command creates a SPOOL DEFINE that specifies attributes for a spooler job:

```
1> ADD DEFINE =MYJOB, CLASS SPOOL, COPIES 3, FORM PAYCHK, &
1> &HOLDAFTER ON, LOC $$.#LLP, OWNER PAYROLL.MANAGER, &
1> &SELPRI 7, REPORT "Nov. Pay"
```

To view all the attributes of this DEFINE, enter:

```
2> INFO DEFINE =MYJOB, DETAIL
  DEFINE NAME           =MYJOB
  CLASS                 SPOOL
  LOC                   $$.#LLP
  COPIES                00003
  FORM                  PAYCHK
  HOLDAFTER             ON
  OWNER                 005,255
  SELPRI                7
  REPORT                NOV. PAY
```

Next, you run the program that produces the spooler output; in this example, the program calculates paycheck amounts from the current payroll records. The OUT file in your command is the SPOOL DEFINE you just created:

```
3> PAYCALC / IN NOV.PAYRCDS, OUT =MYJOB /
```

For detailed information on how to use DEFINES, refer to the *TACL Reference Manual*.

The Spooler and Batch Jobs

A **batch job** is an ordered set of one or more jobs which are linked together and assigned a **batch number** by the spooler. The batch number allows member jobs to be referred to and manipulated as a set. Commands also allow member jobs to be deleted from or added to a batch job.

A batch job is considered open if one or more member jobs are open; that is, they are still being collected. Once a batch job is closed, it is scheduled for delivery in the same way as a single job. The spooler actually schedules just the first job in the batch, but the total lines and total pages attributes are changed to reflect totals for the entire batch.

When a batch job is printed, each member job is delivered to the device in the sequence in which it is linked in the batch. Optional batch headers can be applied to identify the beginning and end of the batch output.

A batch job can be created in the spooler only when a job is opened that originates from a process that has a JOBID attribute. The JOBID attribute may be established using a DEFINE or a RUN parameter. When such a job is opened, a new batch job may be created or the job may be linked to an existing open batch job. To be linked to an existing open batch job, all of the new job's **key attributes** must match those of the open job in the batch job.

The key attributes that determine whether a spooler job is part of a batch job are described in [Table 2-2](#).

Table 2-2. Key Attributes

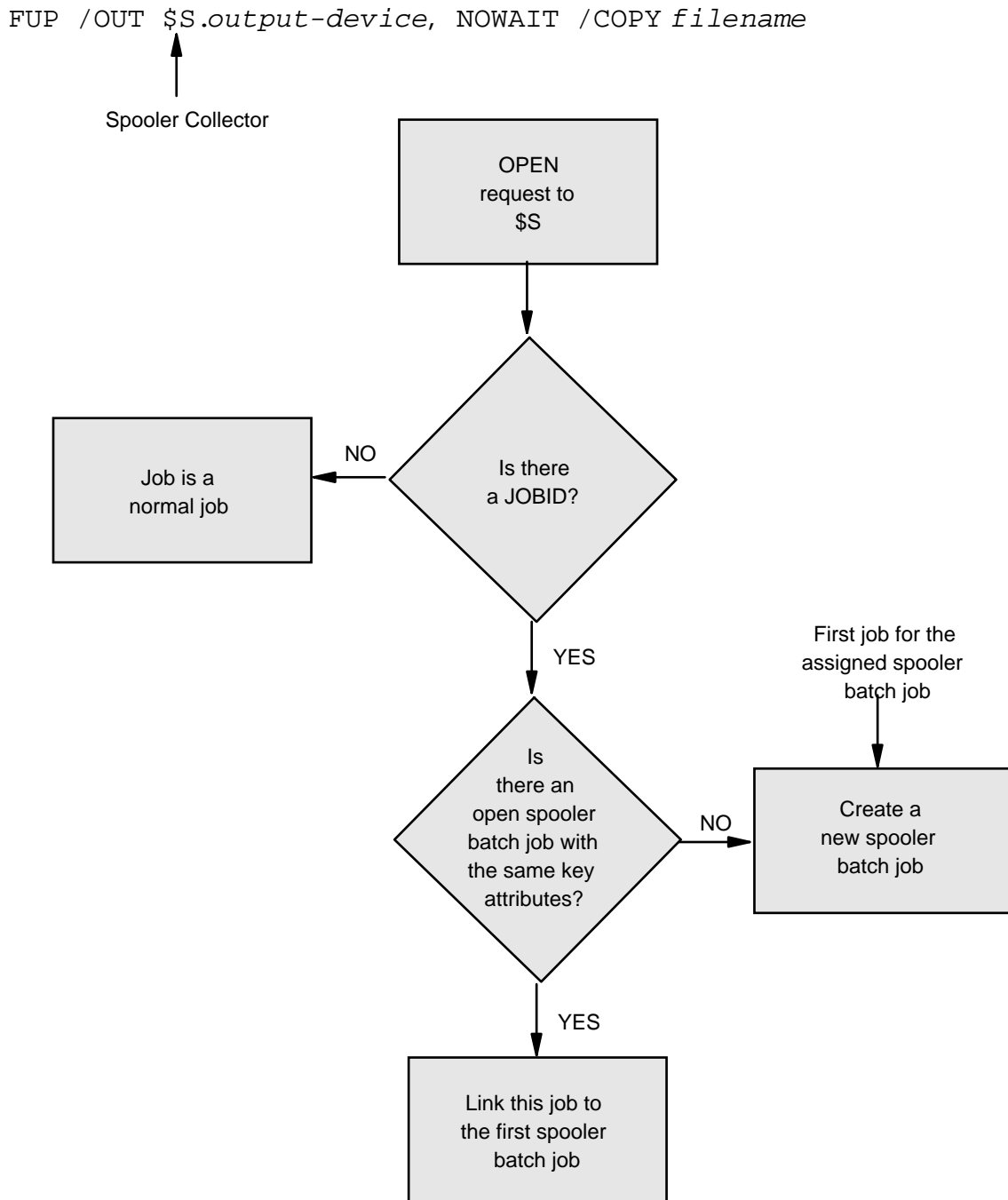
Key Attribute	Description
JOBID	This identifier is the same for all jobs in a batch job. A job that is not part of a batch job does not have this identifier. JOBID can be set through a TACL RUN command or a DEFINE for the application generating the batch job.
FORM	Each job in a batch job must use the same print device and paper.
OWNER	Each job in a batch job must be owned by the same user.
Device where the job is printed	The device where the job is printed must be the same for all members of a batch job. This does <i>not</i> mean that location is a key attribute; jobs can have different locations and still be members of the same batch as long as their locations map to the same device*
GMOM identity of the process that produced the job	This ensures that batches created from different batch executives will be different, even though all of the other key attributes match.

*Locations that do not have a single device mapping (for example, groups or dummy locations) are considered to be mapped to a **null device**. Because locations can be mapped to a null device, jobs whose key attributes and locations map to the null device can be in the same batch job even though their actual locations may vary. Batch jobs that specify the null device are delivered to the location that is established for the first member of the batch job.

Other job attributes have no affect on whether the job is linked to a batch job. Some attributes might have no meaning for a job that is part of a batch job. For example, if every job within a batch job contains a different report name, only the report name associated with the first job in the batch job is printed on the header pages for that batch job. Attributes that have meaning only for the first job in a batch job are the report name, the batch name, and the selection priority.

The method that the spooler uses to determine whether a job becomes a normal job, the first job in a batch job, or a job to be linked to an existing batch job is shown in [Figure 2-6](#).

Figure 2-6. How the Spooler Determines Which Jobs Are to Be Batched



CDT 004CDD

How to Change the Attribute Values of Batch Jobs

With the exception of key attributes, you can change any attribute of a single job that belongs to a batch job. Use Peruse or Spoolcom to change these attributes in the same way you would if the job did not belong to a batch job.

Note. When a job is part of a batch job, you cannot alter the key attributes. Unlinking jobs is described in [Linking and Unlinking Jobs](#) on page 2-22.

You can use Peruse or Spoolcom to change the following attributes for all jobs in a batch job:

- COPIES
- FORM
- HOLD
- HOLDAFTER
- OWNER
- PRI
- REPORT

Peruse commands are described in [Section 3, Peruse](#) and Spoolcom commands are described in [Section 4, Spoolcom](#).

Linking and Unlinking Jobs

Use the Peruse LINK command or the LINK option of the Spoolcom BATCH command to link a job to a batch job if the job's key attributes FORM, OWNER, and LOC match those of the batch job. A job that has not been part of a batch job does not have the key attribute JOBID. When you link a job to the current batch job, it acquires the JOBID of that batch job.

To unlink a specified job from a batch job, use either the Peruse UNLINK command or the Spoolcom BATCH command with the UNLINK option.

For further information about batch jobs, refer to the *NetBatch User's Guide*.

Ensuring Device Uniqueness for Batch Jobs

Applications can use only the first part of a location name—the group name—in their applications and successfully maintain the uniqueness of the batch jobs generated by those applications. However, when the spooler expands location names and must use the default destination name, locations may appear to be the same and batch jobs will share the same batch numbers. To ensure that jobs combine properly into a unique batch job, make sure that the locations used when spooling jobs are precise enough to represent the same device; otherwise, batch jobs may be split because the key attributes do not match when the batch is forming.

Dummy locations represent the same null device key attribute value. In other words, jobs whose output is directed to dummy locations, regardless of their value and preciseness of their locations, will be grouped together in the same batch job if all other key attributes match.

Group locations are also represented by the same null device attribute even though there are devices specified for the group, which allows batch jobs to be formed without direct knowledge of the device that will be used by the group. When printing of the batch job is established for the first member job, all members of the batch are printed to this same device. The total print effort for all of the jobs in a batch job is considered when a device is selected from a group for printing a batch job.

Spooler Security

Each user can have his or her own spooler subsystem to provide for individual secure printing needs. There can be multiple spooler subsystems running at one time on the same system. All command requests to a certain spooler supervisor are validated. If the validation fails, the request is rejected and a SECURITY VIOLATION message is returned to the requester.

When you open a spooler supervisor (using either Peruse or Spoolcom), you will see listed only the jobs that you are authorized to see. You are always authorized to see the jobs that you created. If manager access is enabled for the spooler and you are a group manager (user ID $n,255$), you can see and access all jobs created by members of your group. You must be a system operator (user ID $255,n$) to access all jobs.

Status Requests

All status information requests are granted by the spooler subsystem. All users can obtain the status of any job by using Spoolcom or obtain the status of only their own jobs by using Peruse. If manager access is enabled for the spooler, a group manager can obtain the status of all jobs owned by his or her group.

All Other Requests

All requests other than status requests are validated for access authorization using the following criteria:

- The job creator always has access to his or her own jobs.
- With manager access enabled for the spooler, a group manager (user ID $n,255$) is allowed access to all jobs owned by the group.
- Any system operator (user ID $255,n$) is allowed access at all times.
- The person who created the spooler (by warmstart or coldstart) is also authorized access at all times.

3 Peruse

Peruse is an interactive program that allows you to examine and change the attributes of your job while it is in the spooler subsystem.

Topics described in this section include the following:

- [Peruse Features](#) on page 3-2
- [Running Peruse](#) on page 3-3
- [Peruse Commands](#) on page 3-9

Note. The command displays shown in this section are applicable to D41 and later versions of the spooler supervisor. Refer to the appropriate version of the *Spooler Utilities Reference Manual* for command displays for earlier versions of the spooler supervisor.

Peruse Features

With Peruse you can

- Examine a job (such as a lengthy compiler listing) before printing or deleting it.
- Display a job while it is being spooled.
- Monitor changes in the status of a job.
- Alter job attributes, such as location, number of copies, and report name.
- Print out specific pages or the entire spooled job.
- Copy a job from the spooler queue to an EDIT file or a spooler job file. (Spooler job files are described in this section.)
- Copy a spooler job file to the spooler queue.
- Link a spooler job to a batch job, unlink a spooler job from a batch job, display the status of batch jobs, and change attributes of batch jobs.

This section contains the complete syntax, considerations, and examples for all the Peruse commands. For additional information on batch jobs, see [The Spooler and Batch Jobs](#) on page 2-19. The Peruse error messages are listed in [Appendix B, Peruse Messages](#).

Information on how to use Peruse and the kinds of tasks that can be accomplished with it is presented in the *Guardian User's Guide*.

Running Peruse

You can run Peruse by entering the following command at the TACL prompt:

```
[ \system.] Peruse [ / run-options / ] [ supervisor ]  
[ ; command ] ...
```

\system

specifies the name of a remote system (node) where Peruse is to execute. The default system is your home system.

run-options

are any run options for the command-interpreter RUN command, separated from each other by commas and set off in the command line by slashes (/). See the description of the command-interpreter RUN command in the *TACL Reference Manual* for a complete list of run options.

If you specify OUT *file-name* as a run option, Peruse writes the data to the file in a manner suitable for the file type, process, or device. The *file-name* parameter must be specified; if the file does not exist, Peruse creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

supervisor

is the name of the spooler supervisor that Peruse communicates with. If *supervisor* is omitted, Peruse assumes that \$SPLS is the supervisor.

command

is a Peruse command. Multiple commands can be specified, separated by semicolons, in a list up to 132 characters long. If one or more commands are specified, Peruse executes them and then terminates.

To run Peruse interactively, do not specify commands.

Running Peruse Interactively

To run Peruse interactively, do not specify Peruse commands in the command line when starting Peruse. Peruse displays its program banner, copyright message, and prompt, an underscore (_). An example follows:

```
1> PERUSE
PERUSE - T9101D48 - (30MAR2001) SYSTEM \EAST
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991
```

—

T9101D48 is the product version of Peruse.
 (30MAR2001) is the release date for this version of Peruse.
 \EAST is the system on which it is running.

If you send a job to print on a remote system, you can start Peruse on that system to examine the status of your job. Peruse displays the name of the supervisor process on the remote node, following the copyright message. The following output indicates that your job has finished printing:

```
4> \WEST.PERUSE
PERUSE - T9101D48 - (30MAR2001) SYSTEM \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1989, 1991
SPOOLER SUPERVISOR IS \WEST.$SPLS
```

—

Running Peruse Noninteractively

To run Peruse noninteractively, specify Peruse commands in the command line when you start Peruse. You can enter multiple commands, separated by semicolons, in a list up to 132 characters long. Peruse will execute the commands and then terminate.

The following command causes Peruse to display information on all the jobs you have in the spooler queue and then list the last page of the most recently spooled job before terminating.

```
1> PERUSE; JOB; LIST LAST
```

Initial Job Status Display

If you have jobs in the spooler queue when you start Peruse, they are listed below the program banner as shown in the following example:

```

PERUSE - T9101D48 - (30MAR2001)      SYSTEM \NS
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991

JOB  BATCH  STATE PAGES COPIES PRI HOLD LOCATION      REPORT
1167 555    READY 1      1      4      #BATCH    STEVEW
1289 555    READY 430    1      4      #BATCH    STEVEW
1298      READY 3      1      4      #NOT BATCH STEVEW
1322 555    OPEN   3      4      #NEW DOCUMNT STEVEW
    
```

Each line in the display describes a different job. The column headers are as follows:

- JOB Shows the job number of each job, as assigned by the supervisor. If the current job is a spooler job file, Peruse displays FILE under this header.
- BATCH Is the spooler batch number. If the job is not part of a batch job, this field is blank.
- STATE Shows the status of each job:
 - OPEN The job is still being collected by the spooler.
 - READY Spooler has finished collecting; the job is queued and waiting to print.
 - HOLD The hold-before-printing flag is on, or the hold-after-printing flag is on and the job has printed.
 - PRINT The job is currently printing.
- PAGES Shows the number of pages in each job. Open jobs are still being collected, so the number of pages is not known.
- COPIES Shows the number of copies of the job to be made.
- PRI Shows the priority of the job. When a job is first sent to the spooler, its default priority is 4. The range is 0 through 7, 0 being lowest.

HOLD	Shows the hold status of the job:
A	The hold-after-printing flag is on. The job remains in the queue after it is printed.
B	The hold flag is on but the job cannot be placed in the HOLD state (the job is in the OPEN state). The job will be placed in the HOLD state when it has been collected.
X	For some reason the job is in error (for example, an application abended while spooling the job).
LOCATION	Shows the location of the job. If the location is associated with a print device, the job is queued for printing. If the location is not associated with a print device, the job remains in the queue and you can use Peruse to examine, print, or delete the job. If a spooler job file is the current job, Peruse displays the file name under LOCATION and REPORT.
REPORT	Shows the report name that is printed in the job header message.

You can produce the Peruse display described above at any time during a Peruse session by entering the JOB command at the Peruse prompt.

If you are a group manager (user ID *n,255*), the JOB command display includes all jobs of members of your group if manager access has been enabled for the spooler. See [SHOWGROUP Command](#) on page 3-47 for more information.

Entering Peruse Commands

After displaying any jobs you have in the spooler queue, Peruse displays its prompt, an underscore (_). You can enter one Peruse command per command line, or you can enter several commands on the same line if you separate the commands with semicolons (;). The maximum length of a Peruse command line is 132 characters. Each line is terminated with a RETURN; for example:

```
_J 123
```

```
_DEL
```

```
_EXIT
```

is the same as

```
_J 123; DEL; EXIT
```


Declaring the Current Job

Most Peruse commands affect only the current job. At the beginning of a Peruse session, however, there is no current job. To declare the current job, use one of the following methods:

- Set the current job explicitly by entering a JOB command with the job number. (You can also set the current job by entering only the job number.)
- Set the current job implicitly by entering a Peruse command or by pressing the RETURN key or a function key. The most recently spooled job becomes the current job. For example, if you enter a LIST command, the most recently spooled job becomes the current job and Peruse lists data for that job.

Note. Before you can delete a job using the DEL command, you must declare it the current job by using one of the methods described above.

After you set the current job, Peruse identifies it in subsequent displays with a “J” to the left of the job number. In the following example, job number 1435 is the current job.

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION
REPORT							
1167		READY	1	1	4		#DEFAULT
REPORTS							
J 1435		READY	430	1	4		#HOLD
JANUARY							

—

Displaying a Job

There are three ways to display the current spooler job using Peruse:

- Use the [LIST Command](#), which is described later in this section.
- Press the Return key to display one line. Press and hold down the Return key to scroll the job until you release the key.
- If you have a page-mode terminal, use the function keys to list lines from the current job. [Table 3-1](#) shows the number of lines listed by each function key.

Table 3-1. Number of Lines Listed by the Function Keys

Function Key	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
Lines Listed	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

Using Spooler Job Files

You can also use Peruse with spooler job files. A spooler job file is an unstructured disk file with file code 129. This type of file contains a spooler job, including print data records and formatting control information (CONTROLS, CONTROLBUFs, and SETMODEs).

A spooler job file does not appear in the job display list unless you declare it as the current job by using the JOB command with the file name. The following example shows a job display list with a spooler job file as the current job:

```

JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION REPORT
J FILE          READY 60    1     4     $DISK.MYSVOL.LISTING
1113           READY 100   1     4     #LLPTOY RECEIVE ANN
1234           READY 15    1     4     #HT10  RECEIVE ANN

```

You can use the following Peruse commands with spooler job files:

```

FILES          NUMCOL          STARTCOL
FIND           PAGE            STATUS
JOB            PURGE           VOLUME
LIST           SJFILES

```

The following Peruse commands are not used with spooler job files:

```

COPIES        HOLD            PRI
DEL           HOLDAFTER     REPORT
FORM          LOC

```

If you attempt to execute one of these commands for a spooler job file, Peruse displays the following message:

```
COMMAND NOT SUPPORTED FOR SPOOLER JOB FILES
```

You can run Peruse and access spooler job files even if the spooler supervisor process is not running. However, if the supervisor process is not running and you enter a command that requires this process, Peruse displays the following message:

```
ACCESS TO SPOOL SUPERVISOR PROCESS FAILED, FILE ERROR 016
```

Peruse Commands

[Table 3-2](#) contains a summary of all Peruse commands. Refer to the *Guardian User's Guide* for more information on how Peruse can help you examine your spooled jobs.

Table 3-2. Summary of Peruse Commands (page 1 of 2)

Command	Function
BATCH	Causes the specified batch job to be current.
COPIES	Alters the number of copies for the current job.
DEL	Deletes the current job.
DEV	Displays the status of a device.
EXIT	Terminates the Peruse session.
FC	Allows you to fix and resubmit a Peruse command.
FILES	Displays the names of all files in a specified or default subvolume. Spooler job file names are preceded by a period (.).
FIND	Finds a string in the current job.
FORM	Changes the form name of the current job.
HELP	Displays the syntax and meaning of Peruse commands.
HOLD	Sets the hold flag for the current job.
HOLDAFTER	Sets the hold-after-printing flag for the current job.
JOB	Displays job information or sets the current job.
LINK	Causes the specified spooler job to be linked to the current batch job if the key attributes of the specified job match those of the batch job.
LIST	Lists pages from the current job to the screen, an output device, or a file; copies a spooler job file to the spooler queue.
LOC	Changes the routing location of the current job.
NUMCOL	Sets the number of columns displayed by the LIST command.
OPEN	Specifies a new spooler supervisor.
OWNER	Changes the owner of the current job.
PAGE	Changes and displays the page position of the current job.
PRI	Changes the printing priority of the current job.

Table 3-2. Summary of Peruse Commands (page 2 of 2)

Command	Function
PURGE	Purges one or more disk files.
REPORT	Changes the report name of the current job.
SHOWGROUP	Enables display to a group manager of all jobs that belong to the group.
SJFILES	Displays the names of all spooler job files in a specified or default subvolume. Spooler job file names are preceded by a period (.).
STARTCOL	Sets the first column to be displayed by the LIST command.
STATUS	Monitors and displays the status of jobs in the system.
UNLINK	Dissociates the specified spooler job from the current batch job.
VOLUME	Sets the current default volume and subvolume used by Peruse.

BATCH Command

The BATCH command causes the specified batch job to become current.

```
BATCH batch-number
```

batch-number

is the number of a batch job that you own. Unless you log on as the super ID (255,255), you are not permitted to affect another user's batch job. The batch job specified by *batch-number* becomes current.

Considerations

- Once a batch job is current, most subsequent commands are applied to all the jobs in the batch job.
- The batch job remains current until it is deleted, another batch job is chosen, or a JOB command is issued.

Example

The following Peruse display shows that jobs 1167, 1289 (part of batch 555), and 1200 are all in the READY state:

```
PERUSE - T9101D48 - (30MAR2001) SYSTEM \NS
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991

JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION REPORT
1167 555 READY 1 1 4 #BATCH STEVEW
1200 READY 3 1 4 #ANOTHERFILE STEVEW
1289 555 READY 430 1 4 #CHECK.TACL STEVEW
```

The following BATCH command makes batch 555 current; then the LOC command changes the location where the batch job is to be printed:

```
_BATCH 555
_LOC #LP2X
```

To see the status of the current job, enter a JOB command:

```
_JOB
```

Peruse then displays the following:

```
JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION REPORT
B 1167 555 PRINT 1 1 4 #LP2X STEVEW
1200 READY 3 1 4 #ANOTHERFILE STEVEW
B 1289 555 READY 430 1 4 #LP2X STEVEW
```

After you set the current batch, Peruse identifies jobs in this batch in subsequent displays with a B to the left of the job numbers. In the example above, jobs 1167 and 1289 are in the current batch (555).

COPIES Command

The COPIES command alters the number of copies for the current job. If there is no current job, then the job most recently spooled job becomes the current job.

```
COPIES number-of-copies
```

number-of-copies

is the number of copies of the current job to be printed. The value is a number from 1 through 32767. The default is 1.

Considerations

- When a job is added to the spooler subsystem, the number of copies is 1.
- For each copy you request, a separate header page also prints (if that printer has the header bit turned on).
- If a batch job is current, the COPIES command operates on all jobs that are part of the current batch job.
- The COPIES command is not supported if the current job is a spooler job file.

Example

The following example illustrates how you can obtain two printed copies of a spooled job. The JOB command displays the status of the job, the COPIES command specifies two copies, and the JOB command checks the status of the job again:

```
_JOB
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 1435          READY  30      1        4          #DEFAULT ACCT BEN
_COPIES 2; JOB
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 1435          READY  30      2        4          #DEFAULT ACCT BEN
_
```

DEL Command

The DEL command deletes the current job from the spooler subsystem.

```
DEL
```

Considerations

- Before you can delete a spooler job, you must make it the current job. Refer to [Declaring the Current Job](#) on page 3-7.
- A job that is printing when the DEL command is issued will stop printing and be deleted.
- If the job to be deleted is linked to a batch job, Peruse unlinks the job from the batch job and then deletes it.
- If the job to be deleted is the first job of a batch job, then the second job becomes the new first job.
- If a batch job is current, Peruse deletes all jobs that are part of the current batch job.
- The DEL command is not supported if the current job is a spooler job file.

Examples

1. In this example, the current job is job 52. You can use DEL to delete it from the spooler:

```
_JOB
```

	JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
J	52		HOLD	2	1	4		#LPRMT3	ACCT SUE
	133		READY	5	1	4		#LP	ACCT SUE

```
_DEL
```

```
—
```

Another JOB command verifies that job 52 has been deleted:

```
_JOB
```

	JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
	133		READY	5	1	4		#LP	ACCT SUE

```
—
```

2. If you put a job on hold during an earlier Peruse session, and you later reenter Peruse and attempt to delete it, the message NO CURRENT JOB is displayed:

```
5> PERUSE
```

```
PERUSE - T9101D48 - (30MAR2001) SYSTEM \EAST
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991
```

```
   JOB BATCH STATE  PAGES COPIES PRI HOLD LOCATION REPORT
   456          HOLD    20    1     4      #DEFAULT ACCT SUE
```

```
_DEL
```

```
NO CURRENT JOB
```

You must first make the job the current job by using the JOB command. Then you can delete it from the spooler:

```
_J 456; HOLD OFF; J
```

```
   JOB BATCH STATE  PAGES COPIES PRI HOLD LOCATION REPORT
J 456          PRINT    20    1     4      #DEFAULT ACCT SUE
```

```
_DEL; J
```

```
—
```

No jobs remain in the spooler subsystem.

3. Before you can delete a job, you must make it the current job. When using Peruse noninteractively, you can use the JOB * command to make the most recently spooled job current so you can delete it:

```
9> PERUSE; JOB *;DEL
```


DEV Command

The DEV command displays the status of a specified device and lists the job queue for that device.

```
DEV $device
```

\$device

is the name of a device on the spooler subsystem.

Considerations

Most of the status information is self-explanatory. FORM is the form name, if any, denoting a special print device or paper associated with that printer.

- The jobs in the device queue are listed in the order that they will print.
- The wait time is the estimated total wait time (including time to print) for each job, once the job that is currently printing has completed. The estimated wait time is computed by using the estimated printer speed that was specified in the Spoolcom DEV SPEED parameter. Refer to the Spoolcom [DEV Command](#) on page 4-20 for more information.

If a high-priority job is added to the queue, all the wait times change accordingly.

Example

By requesting the status of device \$LP, you can better determine how long you might have to wait to get a job printed on it:

```
_DEV $LP
```

```
DEV STATE: PRINTING      FORM:

  JOB   OWNER      PAGES  WAIT      FORM
  576   008,005    22     00:05:11
  1324  001,013    40     00:13:36
  344   007,022    21     00:19:11
```

—

EXIT Command

The EXIT command ends the Peruse session.

```
E[XIT]
```

Consideration

Pressing Ctrl-Y also ends the Peruse session.

Example

When you exit Peruse, you receive a command-interpreter prompt:

```
_EXIT
```

```
4 >
```

FC Command

FC (Fix Command) lets you modify and resubmit the last command line entered. The FC subcommands are similar to those used for the command-interpreter FC command, described in the *TACL Reference Manual*.

FC

Consideration

- The FC command executed in Peruse can modify only the last line, unlike an FC command executed at a TACL prompt, where the FC command can also identify and modify other lines.

Example

In this example, the user tried to use the DEV command with #LP instead of \$LP, and received the message ENTRY NOT FOUND. The user can fix the command by replacing # with \$ and reexecuting it:

```
_DEV #LP
```

```
ENTRY NOT FOUND
```

```
_FC
```

```
_DEV #LP
```

```
. $ (press Return)
```

```
_DEV $LP
```

```
. (press Return)
```

```
DEV STATE: PRINTING          FORM:
```

JOB	OWNER	PAGES	WAIT	FORM
1435	009,013	377	00:05:55	

```
—
```

FILES Command

The FILES command displays the names of all files on a subvolume. The names of spooler job files are preceded by a period (.).

```
FILES [ subvol ]
```

subvol

is the subvolume whose file names are to be listed. *subvol* has the form

```
[\system.] [$volume.] [subvolume]
```

If you omit *\system*, PERUSE uses the current default system name.

If you omit *\$volume*, PERUSE uses the current default volume name.

If you omit *subvolume*, PERUSE uses the current default subvolume name. Omit the final period (.) if you specify *\system* or *\$volume* without *subvolume*.

Examples

1. This command lists the names of files on the current default subvolume. Note that LISTING is a spooler job file. This is shown by the period (.) that precedes the name.

```
_FILES
\SYSII.$DISK.SUBVOL
  OBJECT      FILE1      LETTER      .LISTING      SAVEOBJ
—
```

2. In this example, the user provides the volume name, and the system and subvolume names are the current default names:

```
_FILES $BIGVOL
\SYSII.$BIGVOL.SUBVOL
  NAMES      .SAVE      PLANS      PROFILE      SOURCE
—
```

3. In this example, the user provides the system and subvolume names, and the volume name is the current default name:

```
_FILES \NEWSYS.MYSVOL
\NEWSYS.$DISK.MYSVOL
  .LDATA      ERRORS      .RECORDS      .LISTING
—
```

FIND Command

The FIND command locates an occurrence of a specified string in the current job and prints the line containing the string. If there is no current job, then the job most recently spooled becomes the current job.

```
F[IND] [ B[OTH] ] [ / string / ] [ start-position ]
```

B [OTH]

specifies that the FIND command should find and display both uppercase and lowercase occurrences of *string*.

string

is a set of printable ASCII characters set off by two identical separators. You can use quotation marks ("), apostrophes ('), or slashes(/) as separators, but actually any ASCII character can act as the boundary for a string.

Peruse scans the current job, starting at the beginning of the file, for an occurrence of the specified string.

If no string is specified, then Peruse scans the current job for the last *string* given, starting from where the last occurrence of the string was found.

start-position

is the page number from which to begin the search. *start-position* can be specified as

```
[ F | L | * | number ] [ offset ]
```

F is the first page of the job.

L is the last page of the job.

* is the current page of the job.

number is a page number.

offset is specified as [+ *number* | - *number*]

If an offset is specified, it is added to or subtracted from the base (F, L, or *) to define the page.

Peruse supports a page range of 0 through 999,999. If the offset page number accessed is a negative number or greater than 999,999, Peruse returns an error message.

Considerations

- The NUMCOL and STARTCOL commands affect the line displayed by FIND. FIND will find *string* even if *string* is located in an area excluded by a NUMCOL or STARTCOL command, but FIND displays only that area permitted by NUMCOL and STARTCOL.
- If no line is displayed after a FIND command is executed, then there are no occurrences of *string* in the remainder of the job.
- FIND without the BOTH parameter distinguishes between uppercase and lowercase letters, so that searching for BOY, for example, does not find boy. If you want to search for a string that may be in either uppercase or lowercase letters, use the BOTH parameter.
- You cannot use the FIND command to locate TFORM commands embedded in your EDIT file. FIND locates only character strings that appear in the formatted copy.
- If a batch job is current, Peruse scans the first job that is part of the current batch job.

Examples

1. This example shows you how to find errors in your TAL compilation. Once you key in the beginning sequence of the error message, you can reexecute the FIND command without repeating the string, and the command continues to show you the next occurrence of that string:

```
_FIND /**** ERROR/
**** ERROR 49 **** Undeclared Identifier
_F
**** ERROR 27 **** Illegal syntax
```

2. The example below shows how Peruse can find a phrase regardless of whether the characters are uppercase or lowercase:

```
_F B 'NONSTOP'
NonStop system software.
—
```

3. In this example, Peruse begins searching on page 95 of the job for the specified string:

```
_F B /debug/ 95
with the DEBUG subcommand.
—
```

FORM Command

The FORM command alters the form name of the current job. If there is no current job, then the job most recently spooled becomes the current job. The form name of a job denotes the requirement for a special print device or paper associated with that printer.

```
FORM [ form-name ]
```

form-name

is a string of as many as 16 letters, digits, and spaces. If you omit *form-name*, Peruse assigns a form name containing all blanks.

Considerations

- For a job to print on a given device, both the job and the device must have the same form name.
- The assignment of form names to jobs and devices is completely arbitrary, but it is intended to prevent jobs that require special print devices or paper from being printed on the wrong device. You can assign any form name to your job, but only the system operator can change the form name of a device. Use the DEV or JOB STATUS command to check for form-name restrictions on your destination device.
- When you queue a job to the spooler, it is given a form name containing all blanks.
- All form names are automatically shifted to upper case.
- The FORM command is not supported for a single job that is part of a batch job. If a batch job is current, the FORM command operates on all jobs that are part of the batch job.
- The FORM command is not supported if the current job is a spooler job file.

Example

In this example, you want your job to print out on a printer loaded with blank paychecks. You send the job to the spooler, specifying the location where your form paper is loaded, as the DEV command confirms:

```

JOBV   BATCH   STATE   PAGES   COPIES  PRI  HOLD  LOCATION  REPORT
J 1221             READY    1      1458   4           #CHKWRTR BKKPG
JEN
```

```
_DEV $CHKWRTR
```

```
DEV STATE: WAITING          FORM: PAYCHECK
```

```
—
```

Then you specify the same form for your job. The JOB STATUS command confirms this:

```

_FORM PAYCHECK
_J S

JOB  BATCH STATE  PAGES COPIES PRI HOLD LOCATION REPORT
J 1221          READY  1     1458  4           #CHKWRTR BKKPG JEN

JOB  FORM                CLOSE TIME
1221 PAYCHECK            11 DEC 91, 09:50:40

JOB  LOCATION          DEVICE          SEQ   COPY   PAGE
1221 #CHKWRTR.DEFAULT $CHKWRTR          1     1     1458

```

If you send a job to a destination device whose form name does not agree with the job, your job does not print. You would receive no information with the LOC command:

```
_LOC #LP1
```

and you would receive an error message when you checked with the JOB #*location-name* command:

```
_J #PRTR4.DEFAULT
```

```
NO JOBS AVAILABLE
```

In order for your job to print, you must change either the job form name or the job destination so that the form names of the job and the destination device agree.

HELP Command

The HELP command describes the syntax and semantics of Peruse commands.

```
HELP [ / OUT list-file / ] [ command-name | ALL ]
```

OUT *list-file*

is the file, process, or device that receives the help message. If you do not specify OUT *list-file*, output is directed to the OUT file specified when Peruse was started by default (the home terminal).

If OUT *list-file* is specified, Peruse writes the data to the file, process, or device in a manner suitable for it. *list-file* must be specified; if the file does not exist, Peruse creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

command-name

identifies the particular command for which you want an explanation. It can be any Peruse command.

ALL

lists a full explanation of all Peruse commands. If you specify neither *command-name* nor ALL, an abbreviated description of all Peruse commands is given.

Examples

1. To print syntax information about the Peruse JOB command on a printer named \$LP1, enter

```
_HELP /OUT $$.#LP1/ JOB
```

—

2. The following command sends all of the HELP command output to an EDIT file. If file MYHELP does not exist, Peruse creates it.

```
_HELP /OUT MYHELP /
```

—

HOLD Command

The HOLD command sets the hold-before-printing flag on or off for the current job. If there is no current job, then the job most recently spooled becomes the current job.

```
HOLD [ ON | OFF ]
```

ON

sets the hold flag on for the current job. When no argument is specified, ON is assumed.

OFF

sets the hold flag off for the current job.

Considerations

- The HOLD command is not supported if the current job is a spooler job file.
- If a batch job is current (set by the BATCH command), the HOLD command operates on all jobs that are part of the current batch job.
- If the current job is member of a batch job but not the first job of a batch (the log job for a NetBatch job), the HOLD command operates on only the specified job regardless of the state of any other job in the batch. This HOLD status remains in effect even if other member jobs of the batch finish printing and are deleted.

Example

When you enter Peruse in this example, job 75 is already printing:

```
_JOB 75; J
  JOB   BATCH STATE PAGES COPIES PRI HOLD LOCATION REPORT
J 75           PRINT 15    1      4      #LP     ADMIN BILL
 1001           READY 33    1      4      #DEFAULT ADMIN BILL
—
```

The following command places a hold on the job. This causes the job to stop printing and removes it from the print queue:

```
_HOLD;J
  JOB   BATCH STATE PAGES COPIES PRI HOLD LOCATION REPORT
J 75           HOLD 15    1      4      #LP     ADMIN BILL
 1001           READY 33    1      4      #DEFAULT ADMIN BILL
—
```

The following command, removes the hold and the job once again begins printing from page 1:

```
_HOLD OFF;J
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
J 75		PRINT	15	1	4		#LP	ADMIN BILL
1001		READY	33	1	4		#DEFAULT	ADMIN BILL

—

HOLDAFTER Command

The HOLDAFTER command sets the hold-after-printing flag on or off for the current job. If there is no current job, then the job most recently spooled becomes the current job. When the hold-after-printing flag of a job is on and the job has finished printing, the spooler places that job in the HOLD state rather than deleting it from the spooler subsystem.

The HOLD command puts a job in the HOLD state immediately; HOLDAFTER allows it to print before being placed in the HOLD state.

```
HOLDAFTER [ ON | OFF ]
```

ON

sets the hold-after-printing flag on for the current job. ON is the default.

OFF

sets the hold-after-printing flag off for the current job.

Considerations

- If a batch job is current, the HOLDAFTER command operates on all jobs that are part of the current batch job.
- The HOLDAFTER command is not supported if the current job is a spooler job file.

Example

After entering Peruse, you can set the hold-after-printing flag on for a job to keep it in the spooler after it has printed. The A under the HOLD column indicates that the hold-after-printing flag is on:

```
JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
722          READY  3        1        4          #HT1    INVENT  PAT
```

```
_HOLDAFTER; J
```

```
JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 722          PRINT  3        1        4      A    #HT1    INVENT  PAT
```

When the job finishes printing, it enters the hold state and the hold-after-printing flag is still on:

```
JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 722          HOLD  3        1        4      A    #HT1    INVENT  PAT
```

At a later time, you can print another copy of this job without repeating the compiling and spooling time. Remove the hold, and your job enters the print queue:

```
_HOLD OFF; J
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
J 722		PRINT	3	1	4	A	#HT1	INVENT PAT

When the job finishes printing, it again enters the HOLD state. You can do this any number of times. When you no longer have use for the job in the spooler, either delete it or set the hold-after-printing flag off before the last time you print it.

JOB Command

The JOB command sets the status of the current job or displays the status of all jobs that you own. Most of the Peruse commands affect only the current job. The meaning of each entry in the job display is given earlier in this section under [Running Peruse](#).

```
J[OB] [ option ]
```

option

is one of the following:

job-number

file-name

*

S [TATUS]

#*location-name*

job-number

is the number of a job that you own. Unless you log on as the super ID (255,255), you are not permitted to affect another user's job. Using this parameter causes the job whose number you entered to become the current job. You can enter *job-number* alone, omitting J[OB].

file-name

is the name of a spooler job file. Using this parameter causes that file to become the current job and to appear in the job display list. Note that if you want to specify a file named S or STATUS, you must give a more fully qualified file name. Otherwise, Peruse will execute the JOB STATUS command.

*

specifies the job most recently added to the spooler subsystem. Entering an asterisk causes the most recently added job to become the current job.

S [TATUS]

returns an expanded display of the status of the current job. See the description of #*location-name*.

#*location-name*

is the name of a spooler location in the form #*group.destination*. It specifies the job most recently sent to that location.

For more information on spooler locations, refer to the *Guardian User's Guide*.

If you do not specify an option, then the status of all jobs that you own is displayed.

Considerations

- If you are a group manager (user ID $n,255$), the JOB command displays the status of all jobs that belong to your group members if manager access is enabled for the spooler (through the Spoolcom SPOOLER, MGRACCESS command). Assuming manager access is enabled for the spooler, you can enable and disable the display of your group's jobs by using the SHOWGROUP command as described later in this section.
- If you are a system operator (user ID $255,n$), the JOB command displays the status of all jobs in the spooler.
- The JOB command with the STATUS option produces a display such as the following for the current job:

```

JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION REPORT
J 310          READY 2      1      4      #HOLD  MANUALS JIM

JOB  FORM                CLOSE TIME
310                14 JUL 97,13:52:05

JOB  LOCATION            DEVICE          SEQ   COPY   PAGE
310  #HOLD.DEFAULT      72      1      1

```

- The first part is similar to the normal JOB command display described under [Running Peruse](#), earlier in this section.
- The second part has three entries that are displayed only if the job is in one or more device queues. Their meanings are as follows:

JOB	is the job number of the current job.
FORM	is the form name of the current job.
CLOSE TIME	is the date and time the collector finished collecting data from the application. A job still spooling has OPEN as its closing timestamp.

- The third part of the display has six entries that are displayed only for jobs that are on one or more device print queues. Their meanings are as follows:

JOB	is the job number of the current job.
LOCATION	is the complete location name of the job. If the job was routed to a group location (no destination name specified) and there are multiple destination devices defined in this group, then there will be a job entry for each device in the group (see Example 2).
DEVICE	is the device associated with this location.

SEQ	is the job sequence number in the device queue. If the job is printing, its SEQ is PRINT. The next job to print on the device has a sequence number of 1.
COPY	is the number of copies of the job that are to be printed.
PAGE	is the number of pages in the job.

Examples

1. The JOB command gives you information on all jobs you currently own in the spooler, one line for each job:

```
_JOB
```

The following information is displayed:

```

JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 1113          READY  100    1        4          #LLPTOY  RECV ANN
 1234          READY  15     1        4          #HT10    RECV ANN

```

2. If you specify one job, you can obtain further information using the STATUS option:

```
_J 1234;J STATUS
```

The following information is displayed:

```

JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 1234          READY  15     1        4          #HT10    RECV ANN

JOB  FORM                                CLOSE TIME
1234                                14 JUL 97, 10:19:13

JOB  LOCATION                                DEVICE                                SEQ  COPY  PAGE
1234 #HT10.DEFAULT                          $HT10.#HT10                          2    1    15
1234 #HT10.HT10                             $HT10.#HT10                          3    1    15

```

3. If you specify the name of a spooler job file, that file becomes the current job and appears in the job display list:

```
_J $DSK.JVOL.LISTING; J
```

The following information is displayed:

```

JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J FILE          READY  60     1        4          $DSK.JVOL.LISTING
 1113          READY  100    1        4          #LLPTOY  RECV ANN
 1234          READY  15     1        4          #HT10    RECV ANN

```


4. You can declare the current job by entering only the job number. In this example, job 1113 is declared the current job and then the job display list is shown:

```
_1113  
_J
```

	JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
J	1113		READY	100	1	4		#LLPTOY	RECV ANN
	1234		READY	15	1	4		#HT10	RECV ANN

—

LINK Command

The LINK command links the specified job to the current batch job. The key attributes of the specified job must match those of the current batch job.

```
LINK job-number
```

job-number

specifies the job to be linked to the current batch job.

Example

The following Peruse display shows that batch 555 contains two jobs, 1167 and 1289. Note that job 1200 is not currently part of batch 555.

```
PERUSE - T9101D48 - (30MAR2001)      SYSTEM \NS
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  1167 555  READY  1      1      4      #BATCH      STEVEW
  1200      READY  3      1      4      #ANOTHERFILE STEVEW
  1289 555  READY 430    1      4      #CHECK.TACL STEVEW
```

To add job 1200 to batch 555, you must first enter the command to make batch 555 current. Then you enter a LINK command to link job 1200 to the current batch job. Enter a JOB command to verify that job 1200 is linked to batch 555.

```
_ BATCH 555
_ LINK 1200
_ JOB

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  B 1167 555  READY  1      1      4      #BATCH      STEVEW
  B 1200 555  READY  3      1      4      #ANOTHERFILE STEVEW
  B 1289 555  READY 430    1      4      #CHECK.TACL STEVEW
```

Note. The location of job 1200, #ANOTHERFILE, is associated with the same device as #BATCH and #CHECK.TACL. If it were not, the error KEY ATTRIBUTES OF JOB DON'T MATCH THOSE OF BATCH JOB would be returned.

LIST Command

The LIST command displays on your terminal, prints, or writes to a disk file or process all or part of the current job. If there is no current job, then the most recently spooled job becomes the current job.

```
L[IST] [ [ type ] / OUT list-file [ ! ] / ]
      [ page-range [ C ] [ N ] [ O ] , ] ...
```

type

specifies the type of file to be created if *list-file* does not already exist. The format of *type* is

```
[ E[DIT] | S[POOL] | C[OMPRESS] ]
```

E[DIT]

specifies that an EDIT file is to be created if *list-file* does not already exist. If *list-file* exists, then it must be an EDIT file. Characters that cannot be displayed (ASCII characters 0 through 31 and 127 through 255) are not written to *list-file*.

S[POOL]

specifies that a spooler job file is to be created if *list-file* does not exist. If *list-file* exists, then it must be either a spooler job file or a process, which Peruse assumes to be a spooler collector.

C[OMPRESS]

is the same as S[POOL] with the addition of ASCII compression. ASCII compression saves disk space by compressing 8-bit ASCII data into 5-bit units in the spooler job file. If you specify C[OMPRESS] and *list-file* is a process, the command is accepted but ASCII compression is not performed.

OUT *list-file*

identifies the file, process, or device to receive the output. If you specify E[DIT], S[POOL], or C[OMPRESS], then *list-file* is subject to the conditions described above. If you did not specify E[DIT], S[POOL], or C[OMPRESS], then the following apply:

- If *list-file* does not exist, Peruse creates a spooler job file. ASCII compression is not performed.
- If *list-file* does exist, Peruse writes data to it in a manner suitable for the file type, process, or device. *list-file* can be an unstructured file, a relative file, an entry-sequenced file, an EDIT file, a spooler job file, a process, or a device.

If you do not specify `OUT list-file`, output is directed to the `OUT file-name` specified when Peruse was started (the home terminal by default). If you specify `E[DIT]`, `S[POOL]`, or `C[OMPRESS]` without specifying `OUT list-file`, these keywords have no effect.

!

specifies that `list-file` is to be purged, if it exists, and a new list file is to be created before listing begins. If `list-file` exists and you do not specify “!”, new data is appended to any data already in the file if the file is a spooler job file, an EDIT file, or an entry-sequenced file.

page-range

describes the set of pages to be listed to `list-file` and can be

`A[LL] | page [/ page]`

where *page* is

`{ F | L | * | number } [+ number | - number]`

The set of pages can be identified by absolute page numbers or with a base-offset notation. Absolute page numbering begins with the first page as page 1. Base-offset notation uses a base, which can be F (the first page), L (the last page), or * (the current page). An offset is then added to or subtracted from the base to define the page.

The total number of pages you can access ranges from 0 through 999,999.

If you do not specify *page-range*, the last page of the job is listed.

If the offset page number you specify represents a negative-numbered page, Peruse returns an error.

If the page you specify has a number that exceeds the 999,999 page limit, Peruse returns an error.

If the page you specify is greater than the the total number of pages in the job, Peruse accesses the last page of the job.

C

if `list-file` is not a spooler job file or spooler collector, causes formatting-control information (CONTROLS, CONTROLBUFs, and SETMODEs) embedded in the job to be written to `list-file` along with the print data. Normally, this information is suppressed, except when listing to a spooler job file or spooler collector. For details about formatting control, refer to the *Guardian Programmer's Guide*.

N

if `list-file` is a spooler job file or a spooler collector, prevents formatting-control information (CONTROLS, CONTROLBUFs, and SETMODEs) embedded in the job

from being written to *list-file* along with the print data. Normally, this information is included when listing to a spooler job file or a spooler collector. For details about formatting control, refer to the *Guardian Programmer's Guide*.

○

causes the data to be displayed in octal representation rather than in ASCII.

Considerations

- If no current job is defined, the LIST command causes the most recently spooled job to become the current job.
- If you create a file for receiving output from the LIST command, take care that it is large enough. If it is not, the LIST command terminates with file-system error 45 (file is full).
- Pressing the Break key while Peruse is listing a job causes the listing to stop and the Peruse prompt to be displayed.
- Pressing the Return key or one of the function keys displays the job by lines from the current line (actually, from the line following the one most recently listed, or from the first line if none has been listed). Refer to [Table 3-1](#).
- The NUMCOL and STARTCOL commands affect the operation of LIST. Peruse will list only the text in each line from STARTCOL for NUMCOL columns. FIND functions if *string* is located in any area, even that excluded by a NUMCOL or STARTCOL command, but it displays only that area permitted by NUMCOL and STARTCOL.
- If a batch job is current, LIST operates on the first job that is part of the current batch job.

Examples

These examples do not show the pages listed because it would require too much space. However, the effect of each example is discussed in detail.

1. This command lists absolute page 23 in text format and absolute page 30 in octal display format:

```
_LIST 23, 30 0
```

2. This command lists pages 15 through 35, inclusive:

```
_L 15/35
```

3. This command lists the last page of the job:

```
_L
```

4. This command lists the second page from the end of the job:

```
_LIST L-2
```

- The LIST command below shows the first page, the second page, and the next-to-last page in the job. Page 5 is not listed because the current page changes during the LIST command. Also see the [PAGE Command](#) on page 3-42.

```
_P 5
_L F, *, L-1
```

- You can use `/OUT $S.#device/` to print all or only a few pages of an already spooled file. The example below prints page 1 and pages 16 through 20 of the job. Because the destination is a spooler collector, formatting-control information is included by default, so the C parameter is unnecessary. (You would use the N parameter to omit formatting-control information.)

```
_LIST /OUT $S.#LP/ 1, 16/20
```

- Spooler accommodates a maximum of 99,999,999 lines in an EDIT file.

The command below lists the entire job to the disk file `$DISK.SVOL.MYFILE`, which is an EDIT file. If `$DISK.SVOL.MYFILE` already exists, `!` specifies that Peruse should purge it and create a new EDIT file with the same name before listing begins.

```
_LIST E /OUT $DISK.SVOL.MYFILE !/ A
```

- The command below lists pages 1 through 20 of the current job to a spooler job file named `$OLDVOL.SAVE.MYJOB`. Because the `!` parameter is omitted, Peruse appends the listed pages to the end of `$OLDVOL.SAVE.MYJOB` if that file already exists.

```
_LIST S /OUT $OLDVOL.SAVE.MYJOB/ 1/20
```

- You can use Peruse to check a spooled TFORM file for errors before you use `/OUT $S.#device/` to print the pages you want. If you find unacceptable errors, you can rerun the job before printing out any pages. If there are no errors, you can follow the example below to print pages 16 through 19. Because the file is a TFORM file, you must specify the C parameter to retain the formatting-control information.

```
_FIND "***ERROR"
_LIST /OUT $S.#LP/ 16/19 C
```

- You can use Peruse to compress data for more efficient archiving. The command below stores the contents of a spooler job (file code 129) into a spooler job file. It compresses ASCII characters into 5 bits instead of 8. The A specifies all pages and the C causes these pages to retain formatting-control information. The file created can be read later using Peruse or any program that uses PRINTREAD.

```
_LIST C /OUT $DISK.SVOL.ARCH01/ AC
```

To print the archived spooler job file, use the Peruse JOB command to specify the file name. Then use the Peruse LIST command to list the file to a printer. You must specify the pages you want to print, as shown below. If you do not specify the pages you want, only the last page is printed (see Example 3).

```
_JOB SVOL.ARCH01  
_LIST /OUT $S.#PRNT3/ 1/27
```

LOC Command

The LOC command alters the location of the current job. If there is no current job, then the job most recently spooled becomes the current job.

```
LOC [ #location-name ]
```

#*location-name*

is the name of the new location for the current job. If #*location-name* is not specified, then the current job is given #DEFAULT (your default printer) as its new location.

Considerations

- The LOC command is not supported for a single job that is part of a batch job. If a batch job is current, the LOC command operates on all jobs that are part of the current batch job.
- The LOC command is not supported if the current job is a spooler job file.

Example

In this example, you have specified #HOLD as *list-file* in order to examine the job before printing:

```
_JOB
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 777                READY  22      1        4        #HOLD  ACCOUNT ED
-
```

Once you finish inspection, use the LOC command to redirect the output to the specified printer for a hard copy:

```
_LOC #LP3;J
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 777                READY  22      1        4        #LP3   ACCOUNT ED
-
```


NUMCOL Command

The NUMCOL command alters the number of columns displayed when a job is listed.

```
NUMCOL number-of-columns
```

number-of-columns

is the number of columns, counting from the left margin, that are to be displayed or printed.

Considerations

- When Peruse is first run, the value for NUMCOL is 0 and the width of the printing device determines the number of columns to be printed. If NUMCOL is set to a value other than 0, the value specified is used as the output width. An even value is recommended. NUMCOL cannot be set to a value less than zero.
- NUMCOL can be used with STARTCOL to list only a few columns of a job.

Example

In the example below, you display lines of your spooled file on your page-mode terminal by pressing the F2 function key, which displays four lines:

_(Press F2 function key)

```
54.      000000 0  0  STRUCT buffer ;
55.      000000 0  0  BEGIN
56.      000000 0  1      INT status,
57.      000000 0  1      reserved;
```

—

Using the NUMCOL command, you can stop the display of all columns to the right of the eighteenth column. You can redisplay the same lines by using the PAGE command with the * parameter before pressing the same function key.

_ NUMCOL 18

_ P*

_(Press the F2 function key)

```
54.      000000 0  0
55.      000000 0  0
56.      000000 0  1
57.      000000 0  1
```

—

OPEN Command

The OPEN command allows you to specify a new spooler supervisor without exiting Peruse.

```
OPEN [ supervisor-name ]
```

supervisor-name

is the name of the new spooler supervisor that Peruse is to communicate with. The process name of the supervisor can be in local or network form.

The local form is `$process-name`, which identifies the spooler process (see [Considerations](#)).

The network form is `\system.$process-name`, where *system* identifies a remote node. (See [Considerations](#) for additional information about local and network names.)

The default spooler supervisor is `$SPLS`.

Considerations

- To open the local supervisor, the supervisor name can be omitted if it is `$SPLS`.
- To open a remote supervisor, the system name can be specified without the process name if the process is `$SPLS`.
- After you execute an OPEN command, Peruse displays the new spooler supervisor name and then lists the jobs that you have in that spooler queue.

Example

You can use the OPEN command to inspect jobs you have spooled to a remote system without exiting Peruse and logging onto that remote system:

```
_ OPEN \FARSYS
```

```
SPOOLER SUPERVISOR IS \FARSYS.$SPLS
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
	1167	READY	1	1	4		#DEFAULT	REPORTS
J	1435	READY	430	1	4		#HOLD	JANUARY

```
—
```

OWNER Command

The OWNER command changes the owner of the current job. If there is no current job, then the job most recently spooled becomes the current job. After this command is issued, the current job becomes the property of the specified owner and can no longer be accessed by the previous owner.

```
OWNER { group-name.user-name | group-# , user-# }
```

group-name.user-name

is the name of the new owner, entered without spaces on either side of the period.

group-# , user-#

is the user number of the new owner.

Considerations

- The default owner of a job is the user who made the initial request to the spooler.
- If a batch job is current, the OWNER command operates on all jobs that are part of the current batch job.

Examples

1. An example of the OWNER command using the *group-name.user-name* form is

```
_OWNER ADMIN.BILL
```
2. An example of the OWNER command using the *group-# , user-#* form is

```
_OWNER 7, 10
```
3. You have a hold on job 454 and want to transfer its ownership to user 7,10. You can do this with the OWNER command. When you next display your jobs, job 454 no longer shows, because it now belongs to user 7,10:

```
_JOB
      JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
      777          READY  2       1       4          #LPRMT3  ACCT JUD
J 454          HOLD   5       1       4          #LPS     ACCT JUD
_OWNER 7,10;J

      JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
      777          READY  2       1       4          #LPRMT3  ACCT JUD
```

PAGE Command

The PAGE command displays or changes the page and line-number position of the current job. If there is no current job, then the job most recently spooled becomes the current job. Page numbers begin with 1 and do not include the header page, if any. Peruse assumes 60 lines to a page.

```
P[AGE] [ number | F | L | * ]
```

number

is the page number of the new page position for the current job.

F

indicates that the new page position for the current job is the first page.

L

indicates that the new page position for the current job is the last page.

*

indicates that the new page position for the current job is the first line of the current page.

When you specify no argument, the spooler displays the page and line number position of the current job.

Considerations

- The current line can also be changed using a FIND command or by listing lines.
- The pages of a spooled job include not only the pages of your spooled file but also any error pages generated by your application. You can use this fact to find your TFORM errors.
- The number of pages supported by the Page command ranges from 0 through 999,999. For a page number greater than 999,999, Peruse returns an error.

Refer to information on using Peruse with TFORM in the *Guardian User's Guide*.

Examples

1. In the following example, the PAGE command tells you the line and page number of where you are located in the spooled job:

```
— PAGE  
— PAGE: 7          LINE: 15  
—
```

2. If you use the * parameter, you are repositioned to the top of the current page:

```
— P *;P  
— PAGE: 7          LINE: 1  
—
```

PRI Command

The PRI command alters the priority of the current job. If there is no current job, then the job most recently spooled becomes the current job. The higher the priority, the sooner a job will print.

```
PRI priority
```

priority

is a number in the range 0 through 7. Zero is the lowest priority. The default priority is 4.

Considerations

- When a job is added to the spooler, it is given a priority of 4.
- The actual queuing algorithm used by the spooler depends on the setting of the device first-in, first-out (FIFO) switch. However, regardless of the FIFO switch, higher-priority jobs are always printed before lower-priority jobs.
 - If the FIFO switch is on, jobs of the same priority are handled on a first-come, first-served basis.
 - If the FIFO switch is off, shorter jobs of the same priority are handled before longer jobs of that priority. However, a long job does not wait indefinitely for shorter jobs.
- If a batch job is current, the PRI command operates on all jobs that are part of the current batch job.
- The PRI command is not supported if the current job is a spooler job file.

For a more complete description of the spooler queuing algorithm, see the *Guardian System Operations Guide*.

Example

In this example, your current job has the default priority of 4. You can change this to 5 with the PRI command, putting your job ahead of other jobs in the queue that have a priority of 4 or lower.

```
_J
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 52          READY  7       1       4          #LP      ACCOUNT ED
_PRI 5;J

  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 52          READY  7       1       5          #LP      ACCOUNT ED
-
```

PURGE Command

The PURGE command deletes one or more disk files. This command applies only to Enscribe files.

```
PURGE file-name [ , file-name ] ...
```

file-name

is the name of a disk file to be deleted. Peruse expands a partial file name by adding the current default names for system, volume, and subvolume. Multiple *file-name* parameters can be specified, separated by commas.

Example

This command purges the file `BADFILE`, after using the current default names to expand the file name to `\SYS22.$MYVOL.GOOD.BADFILE`.

```
_PURGE BADFILE  
  
\SYS22.$MYVOL.GOOD.BADFILE PURGED  
—
```

REPORT Command

The REPORT command alters the report name of the current job. If there is no current job, then the job most recently spooled becomes the current job. The report name is usually printed in the header in large banner-sized letters at the beginning of the job.

```
REPORT [ report-name ]
```

report-name

is the new report name for the current job. *report-name* is composed of as many as 16 letters, digits, and blanks, and must begin with a letter.

If *report-name* is omitted, then the current job is assigned a report name containing all blanks.

In the case of the print processes supplied by Tandem, the first 8 characters of *report-name* are used as the first line of the banner heading characters on the header page; characters 9 through 16 appear on the second line.

Considerations

- Normally, when a job is collected by the spooler, its *report-name* is the same as the group and user name of the owner. A different *report-name* (other than the owner name) can be specified when the job is collected by means of parameters supplied to the SPOOLSTART or SPOOLEND procedures. Refer to the *Spooler Programmer's Guide* for exceptions.
- Report names are automatically shifted to uppercase letters.
- If a batch job is current, the REPORT command operates on all jobs that are part of the current batch job.
- The REPORT command is not supported if the current job is a spooler job file.

Example

After sending your job to the spooler, you want to change the report name that will appear in the header message. Notice that you must insert spaces so that the two-name header is properly printed on two lines.

```

__JOB
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 123          HOLD    2       1       4          #HOLD    ACCNT   JOHN

__REPORT NEW      TEST1
__JOB
  JOB  BATCH  STATE  PAGES  COPIES  PRI  HOLD  LOCATION  REPORT
J 123          HOLD    2       1       4          #HOLD    NEW     TEST1

```


SHOWGROUP Command

The SHOWGROUP command enables or disables the display to a group manager (user ID *n,255*) of all jobs that belong to group members. This command affects the displays produced by the JOB and STATUS commands.

```
{ SHOWGROUP | SG } [ ON | OFF ]
```

ON

enables display to a group manager of all the jobs that belong to the manager's group members. This is the default value and is equivalent to entering SHOWGROUP.

OFF

disables display to a group manager of all jobs that belong to the group.

Considerations

- Before SHOWGROUP ON can be enabled, group manager access must be enabled for the spooler by setting MGRACCESS ON with the Spoolcom SPOOLER command. Only an authorized user can enable group manager access. See the SPOOLER command description in Section 4, Spoolcom.
- Assuming group manager access is enabled for the spooler and you, a group manager, invoke Peruse, all jobs belonging to your group are displayed in addition to your own. You can then access all the jobs as if you were the owner. For example, if your group number were 10, your user ID would be 10,255 and you could display and access all jobs for users with a user ID of 10,*n*.
- Assuming group manager access is enabled for the spooler and you have set SHOWGROUP OFF, you can still access jobs belonging to your group even though they are not displayed by the JOB and STATUS commands.
- If manager access is disabled for the spooler, the SHOWGROUP command has no effect.
- If MGRACCESS is set off while you are running a Peruse session, the current job is accessible even if it belongs to a group member. You need to reestablish the current job.

Examples

1. The following example shows the initial JOB command display for a group manager invoking Peruse when manager access is enabled for the spooler. Only job 3080 (report name AD MGR) actually belongs to the manager.

```
4> PERUSE
PERUSE - T9101D48 - (30MAR2001) SYSTEM \FORTY
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
3074		HOLD	10	1	4		#LASER1	AD LEE
3076		READY	7	1	4		#LASER3	AD ROBIN
3080		HOLD	4	1	4		#LASER1	AD MGR
3084		READY	55	1	4		#LASER1	AD LEE

2. The following example shows the group manager changing the status of job 3074. Then the group manager enters the JOB command for another display of status, which shows job 3074 with the READY status.

```
_JOB 3074;HOLD OFF
_JOB
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
J 3074		READY	10	1	4		#LASER1	AD LEE
3076		READY	7	1	4		#LASER3	AD ROBIN
3080		HOLD	4	1	4		#LASER1	AD MGR
3084		READY	55	1	4		#LASER1	AD LEE

3. The following example shows the group manager setting SHOWGROUP OFF and getting a display of his or her own jobs.

```
_SHOWGROUP OFF
_JOB
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
3080		HOLD	4	1	4		#LASER1	AD MGR

4. The following example shows the Spoolcom SPOOLER STATUS command you can use to determine whether manager access is enabled. The display shows whether MGRACCESS is ON or OFF.

```
5> FontRPSetup;SPOOLER,STATUS DETAIL
```

```
SPOOL SUPERVISOR: $SPLS
STATE: ACTIVE
MGRACCESS: ON
LOG FILE: $0
LAST ERROR: NONE
```

SJFILES Command

The SJFILES command displays the names of all spooler job files on a subvolume. The names of spooler job files are preceded by a period (.).

```
SJFILES [ subvol ]
```

subvol

is the subvolume whose file names are to be listed. *subvol* has the form:

```
[\system.] [$volume.] [subvolume]
```

If you omit *\system*, Peruse uses the current default system name.

If you omit *\$volume*, Peruse uses the current default volume name.

If you omit *subvolume*, Peruse uses the current default subvolume name. Omit the final period (.) if you specify *\system* or *\$volume* without *subvolume*.

Examples

1. This command lists the names of spooler job files on the current default subvolume:

```
_SJFILES
\SYSTEM.$DISK.SUBVOL
.LPROG      .LISTING      .TEXT
—
```

2. In this example, the volume name is provided while the system and subvolume names are the current default names:

```
_SJFILES $BIGVOL
\SYSTEM.$BIGVOL.SUBVOL
.SAVEDATA  .FILE1      .OUTFILE
—
```

3. In this example, the system and subvolume names are provided while the volume name is the current default name:

```
_SJFILES \NEWSYS.MYSVOL
\NEWSYS.$DISK.MYSVOL
.LDATA     .RECORDS    .OLDLIST
—
```

STARTCOL Command

The STARTCOL command alters the starting column number used when a job is listed. The first column is 1.

```
STARTCOL starting-column
```

starting-column

is a number greater than 0 and less than or equal to the rightmost column in the current job. The default column number is 1.

Considerations

- When Peruse is first run, the value for STARTCOL is 1.
- STARTCOL can be used with NUMCOL to list only a few columns of a job.
- You can use STARTCOL to view those lines of a report that are longer than 80 characters and that otherwise cannot be seen on your terminal.

Example

In this example, you use the F2 function key to display four lines of your program on your page-mode terminal:

_(Press the F2 function key)

```
54.      000000 0 0  STRUCT buffer ;
55.      000000 0 0  BEGIN
56.      000000 0 1      INT status,
57.      000000 0 1      reserved;
```

—

Then you use the PAGE * command to reposition yourself to the first line of the page and execute a STARTCOL command to suppress the first 18 columns of the program. This time the F2 key displays only the instruction portion of your program:

_PAGE * (reposition to first line of the page)

_STARTCOL 18

_(Press the F2 function key)

```
STRUCT buffer ;
BEGIN
      INT status,
      reserved;
```

—

STATUS Command

The STATUS command displays the status of all available jobs each time the state of any of the jobs changes. A C in front of a status line means that the state of that job has changed since the last status display.

```
S [STATUS] [ delay ]
```

delay

is the number of seconds to wait between each status check. The range of values permitted for this argument is 2 through 32767. When no delay is specified, Peruse waits 10 seconds between status checks.

Considerations

- The jobs displayed by the STATUS command depend on your user authorization as follows:
 - If you are a typical, unauthorized user, the STATUS command displays only your own jobs.
 - For a group manager (user ID *n,255*), the STATUS command displays the status of all jobs that belong to group members if manager access is enabled for the spooler (through the Spoolcom SPOOLER, MGRACCESS command). Assuming manager access is enabled for the spooler, you can enable and disable the display of your group's jobs by using the [SHOWGROUP Command](#).
- Immediately before a changed display is given, the spooler sends a CTRL/G to the terminal. This causes the terminal to issue an alert sound.
- A C in a column to the far left indicates those jobs that have changed status since the last display.
- STATUS is a continuous display; to exit STATUS and display the Peruse prompt, press the Break key.

Example

After sending job 639 to the spooler location \$S.#HOLD, using the NOWAIT option, you enter Peruse. When you execute the STATUS command, a status display appears, after which the Peruse prompt does not return; Peruse is waiting for the status of a job to change.

```
_STATUS
```

	JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
	633		READY	7	1	4		#LP	PAYROLL
J	639		OPEN		1	4		#HOLD	PAYROLL

The status of a job has changed: job 639 has completed spooling and now has a page count. The terminal beeps and Peruse displays the status information. The C indicates that this job has changed status since the last display.

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
633		READY	7	1	4		#LP	PAYROLL
JC639		READY	5	1	4		#HOLD	PAYROLL

Job 633 is ahead of job 639 in the printer queue and job 633 now begins printing. The C moves to indicate the most recent status change:

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
C633		PRINT	7	1	4		#LP	PAYROLL
J 639		READY	5	1	4		#HOLD	PAYROLL

Job 633 finishes printing and no longer appears on the status display. Job 639 remains at location #HOLD. To display the Peruse prompt, press the Break key.

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
J 639		READY	5	1	4		#HOLD	PAYROLL

(Press the Break key now)

—

UNLINK Command

The UNLINK command unlinks the specified job from the current batch job.

```
UNLINK job-number
```

job-number

specifies the job to be unlinked from the current batch job.

Example

The following Peruse display shows three jobs (1167, 1200, and 1289) in batch 555:

```
PERUSE - T9101D48 - (30MAR2001) SYSTEM \NS
Copyright Tandem Computers Incorporated 1978, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1991
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
1167	555	READY	1	1	4		#BATCH	STEVEW
1200	555	READY	3	1	4		#ANOTHERFILE	STEVEW
1289	555	READY	430	1	4		#CHECK TACL	STEVEW

To unlink job 1200 from batch 555, first enter the BATCH command to make batch 555 current, followed by the UNLINK command. To verify that job 1200 is unlinked from batch 555, enter a JOB command:

```
_BATCH 555
_UNLINK 1200
_JOB
```

JOB	BATCH	STATE	PAGES	COPIES	PRI	HOLD	LOCATION	REPORT
1167	555	READY	1	1	4		#BATCH	STEVEW
1200		READY	3	1	4		#ANOTHERFILE	STEVEW
1289	555	READY	430	1	4		#CHECK TACL	STEVEW

VOLUME Command

The VOLUME command changes the current default volume or subvolume names, or both, used with Peruse. The initial default names are the subvolume and volume names that were the current default names when you started Peruse.

```
VOLUME [ [ \system. ] $volume. subvol ]
        [           $volume           ]
        [                   subvol     ]
```

\system

sets the current default system name for Peruse. You can set the default system name only in a named system, such as one in a network.

\$volume

sets the current default volume name for Peruse.

subvol

sets the current default subvolume name for Peruse.

Considerations

- Entering VOLUME with no volume or subvolume name restores the default names that were in effect when you started Peruse.
- Peruse and the command interpreters have separate default names for system, volume, and subvolume. When you return to the command-interpreter prompt from Peruse, the default names are reset to the values that were in effect before you started Peruse.

Examples

1. This command sets the current default subvolume name for Peruse to SUBVOL1 but does not change the current default volume or system:

```
_VOLUME SUBVOL1
```

2. This command sets the current default volume name for Peruse to \$DEFVOL but does not change the current default subvolume or system:

```
_VOLUME $DEFVOL
```

3. This command sets the current default system, volume, and subvolume names for Peruse to \ITALY.\$MILANO.ARTWORK:

```
_VOLUME \ITALY.$MILANO.ARTWORK
```

4 Spoolcom

Spoolcom is an interactive program that gives you control of the spooler. This section contains complete syntax, considerations, and examples of Spoolcom commands for authorized users and others.

Topics described in this section include the following:

- [Spoolcom Features](#) on page 4-2
- [Spoolcom Security](#) on page 4-2
- [Running Spoolcom](#) on page 4-2
- [Spoolcom Commands](#) on page 4-5

Information on how to use Spoolcom interactively and noninteractively is described in the *Guardian User's Guide*. Spoolcom error messages appear in [Appendix C, Spoolcom Messages](#).

Note. The command displays shown in this section are applicable to D41 and later versions of the spooler supervisor. Refer to the appropriate version of the *Spooler Utilities Reference Manual* for command displays for earlier versions of the spooler supervisor.

Spoolcom Features

With Spoolcom, you can

- Display the status of collectors, devices, jobs, print processes, routing structures, and the spooler itself
- Change the location, state, or any attribute of your job
- Delete your job from the spooler subsystem
- Restart a device that has gone offline with a device error

Other operations performed using Spoolcom can effect the entire spooler subsystem. For this reason, only authorized users can perform these tasks, which are described in the *Guardian User's Guide*.

Spoolcom Security

Each user can start his or her own spooler process. The user starting the individual spooler process is called the creator. Authorization to access the spooler process or to change its configuration parameters is limited, depending on the user ID of the creator of the spooler process.

- The job creator always has access to his or her own jobs.
- Any system operator (user ID 255, *n*) is allowed access at all times.
- The person who created the spooler process (by warmstart or by coldstart) is also authorized access at all times.

All status information requests are granted by the spooler subsystem. Any user can obtain the status of any job by using Spoolcom or can obtain the status of only his or her own jobs by using Peruse.

Running Spoolcom

You can run Spoolcom by entering the following command at the TACL prompt:

```
SPOOLCOM [ / run-options / ] [ supervisor ] [ ; command ] ...
```

run-options

are any run options for the command-interpreter RUN command, separated from each other by commas. See the description of the command-interpreter RUN command in the *TACL Reference Manual* for a complete set of the run options.

If you specify *OUT filename* as a run option, Spoolcom writes the data to the file in a manner suitable for the file type, process, or device. *file-name* must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101).

The file can be an EDIT file, unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

supervisor

is the name of the spooler supervisor with which Spoolcom communicates. If *supervisor* is omitted, then Spoolcom assumes \$SPLS is the supervisor.

If you want to communicate with a remote supervisor, you should use the network form of the supervisor process name (for example, \FAR.\$SPLS) rather than invoking the remote Spoolcom (for example, \FAR.SPOOLCOM).

command

is a Spoolcom command. You can specify multiple commands, separated by semicolons, in a list up to 132 characters long. If one or more commands are specified, Spoolcom executes them and terminates.

Running Spoolcom Noninteractively

To run Spoolcom noninteractively, specify Spoolcom commands in the command line when starting Spoolcom. You can enter multiple commands, separated by semicolons, in a list up to 132 characters long. The maximum length of the Spoolcom command line is 132 characters, regardless of whether it is entered from a terminal or read from a disk file. The following command causes Spoolcom to obtain the status of job 43 and then exit:

```
)JOB 43, STATUS; EXIT
```

Running Spoolcom Interactively

To run Spoolcom interactively, do not specify Spoolcom commands in the command line when starting Spoolcom. Spoolcom displays its program banner, copyright message, and the Spoolcom prompt, which is the close parenthesis character, “)”.:

```
SPOOLCOM - T9101D48 - (25JUN2001)    SYSTEM \EAST
Copyright Tandem Computers Incorporated 1997
)
```

T9101D48 is the version of Spoolcom.

(25JUN2001) is the release date for this version of Spoolcom.

\EAST is the system Spoolcom is running on.

Entering Spoolcom Commands

A Spoolcom command consists of a command word sometimes accompanied by a parameter; it can be followed by one or more subcommands. The command and its parameter are separated from the subcommands by commas. Subcommands are also separated from each other by commas.

For example, to specify the report name `TAL_COMPILE` for job number 1635, enter the following Spoolcom command:

```
)JOB 1635, HOLD, REPORT TAL_COMPILE, START
```

`JOB` is the command, and the parameter 1635 indicates that job number 1635 is being referred to. To change the job report name, you must put the job in the `HOLD` state. Use the subcommand `REPORT` to specify the report name `TAL_COMPILE` for the job. (The spooler converts all keyed input to uppercase letters.) After changing the report name, use the subcommand `START` to put the job back in the device queue. (In the comparable statement in *Peruse*, the `HOLD` and `START` of a job are invisible to the user.)

Spoolcom allows you to enter commands affecting a job, a collector, or any other spooler component on a single line or on separate lines. However, each command line must be complete. For example, to enter the above subcommands on three separate lines, you must repeat the command `JOB` and the parameter 1635 on each line:

```
)JOB 1635, HOLD  
)JOB 1635, REPORT TAL_COMPILE  
)JOB 1635, START
```

These three commands have the same effect as the single command line shown earlier. In fact, each subcommand can be viewed as a separate command. Each subcommand is processed left to right, with each being completely processed before the next subcommand is executed. The only exception to this is the `DRAIN` subcommand, because it can take some time for a component to completely drain. Spoolcom puts the component in the drain state but does not wait for the drain to finish.

Spoolcom Commands

A Spoolcom command summary describing the features available to all users is given in [Table 4-1](#).

Table 4-1. Spoolcom Command Summary for All Users

Command	Function
BATCH	Alters attributes and changes the state of your batch jobs. This command also obtains the status of any batch job in the spooler subsystem.
COLLECT	Obtains the status of spooler collectors.
COMMENT	Designates a comment to be ignored by Spoolcom.
DEV	Controls and obtains the status of devices in the spooler subsystem. While a device is printing a job, the owner of that job can issue a SKIP or SKIPTO subcommand on that device. Anyone can use START on an offline device that has not been put offline by a DRAIN subcommand issued.
EXIT	Terminates a Spoolcom session.
FC	Allows edit and reexecution of the previous command line. It is similar to the TAACL FC command, but works only for the most recent previous command.
HELP	Displays the syntax of Spoolcom commands for user reference.
JOB	Alters attributes and changes the state of your jobs. This command also obtains the status of any jobs in the spooler subsystem.
LOC	Displays the status of the spooler routing structure.
OPEN	Specifies the particular spooler subsystem to which the other Spoolcom commands refer.
PRINT	Obtains the status of the spooler print processes.
SPOOLER	Obtains the status of the spooler.

[Table 4-2](#) is a Spoolcom command summary describing the features available only to authorized users. Note the differing functions of these commands, depending on the authorization of the user.

Table 4-2. Spoolcom Command Summary for Authorized Users

Command	Function
BATCH	Specifies attributes, obtains status, and changes the state of any batch job in the spooler subsystem.
COLLECT	Specifies attributes, obtains status, and changes the state of collectors.
COMMENT	Designates a comment to be ignored by Spoolcom.
DEV	Specifies attributes, obtains status, and changes the state of devices.
EXIT	Terminates a Spoolcom session.
FC	Allows edit and reexecution of a command line. It is similar to the TACL FC command.
FONT	Causes the spooler to recognize a job as a font job and to associate that job with a specified name. A font job contains commands that can be sent to a programmable printer.
HELP	Displays the syntax of Spoolcom commands for user reference.
JOB	Specifies attributes, obtains status, and changes the state of jobs.
LOC	Defines and modifies the spooler routing structure.
OPEN	Specifies the particular spooler subsystem to which the other Spoolcom commands refer.
PRINT	Specifies attributes, obtains status, and changes the state of print processes.
SPOOLER	Starts, stops, and obtains the status of the spooler; obtains a dump of the supervisor process; and sets group manager access.

BATCH Command

Authorized users can use the BATCH command to specify attributes, obtain the status, and change the status of any batch job.

Refer to the *Guardian User's Guide* for a description of batch job states and default attributes.

All users can perform these operations on their own batch jobs.

```
BATCH [ batch-number ] [ , subcommand ] ...
```

batch-number

is the number of the batch job.

subcommand

is one of the subcommands described below. If *subcommand* is not specified, then STATUS is assumed. Unauthorized users can use these subcommands to change the attributes and status of only their own batch jobs.

COPIES *number-of-copies*

specifies the number of copies to be printed. A job must be in the HOLD state before this attribute can be changed. The range of values is 1 through 32767. The default value is 1.

DELETE [!]

deletes all the jobs associated with the specified batch number. If you enter a JOB DELETE command on a job that is currently printing, Spoolcom might indicate that the job has been deleted before it actually is, because the supervisor can delete the job only after the print process stops printing the job. Enter DELETE ! to have the print process stop printing the job and delete all jobs associated with the specified batch number.

FORM [*form-name*]

specifies a form name for the batch job. A batch job must be in the HOLD state before its form name can be changed. A form name is composed of up to 16 letters, numbers, and blanks. The default form name is all blanks.

HOLD

places the batch job in the HOLD state. A batch job must be in the HOLD state before most other attributes can be modified. To remove a batch job from the HOLD state, use the START subcommand.

HOLDAFTER [OFF | ON]

sets the hold-after-printing flag on or off. HOLDAFTER or HOLDAFTER ON places the batch job in the hold state after it is printed. HOLDAFTER OFF (the default) allows the batch job to be deleted after printing.

LINK *job-number*

causes the job specified by *job-number* to become linked to the current batch job if the key attributes of the specified job match those of the batch job.

LOC [#*location*]

specifies a new location for the batch job. The batch job must be in the hold state before its location attribute can be changed. If the #*location* parameter is omitted, #DEFAULT is the new location.

OWNER { *group-name.user-name* | *group-number,user-number* }

designates a new owner for the batch job. A batch job must be in the HOLD state before its owner can be changed. You must use the local form of either the group and user name or the group and user number. The range for *group-number* and *user-number* is 1 through 255.

REPORT [*report-name*]

changes the report name for the batch job. A batch job must be in the hold state before its report name can be changed. The *report-name* is composed of as many as 16 letters, numbers, and blanks, and it must begin with a letter. It prints on the header page, 8 characters to a line. The default report name is the user ID of the creator, that is, *group-name.user-name*.

SELPRI *selection-priority*

specifies the selection priority of the batch job. A batch job must be in the HOLD state before its priority attribute can be changed. Valid priority values are 0 through 7. When a batch job is added to the spooler, it is given a default priority of 4. The higher the priority, the sooner the batch job prints.

START

takes a batch job out of the hold state and places it in the ready state.

STATUS [/ OUT *file-name* /][DETAIL]

obtains the status of all batch jobs in the spooler subsystem.

OUT *file-name*

indicates where Spoolcom is to write the batch job status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If `OUT filename` is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. *file-name* must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

DETAIL

requests a complete list of batch job attributes. This parameter displays the attributes of batch jobs in the spooler subsystem. If *batch-number* is not specified, the status of all batch jobs in the spooler subsystem is returned. If *batch-number* is specified, the status of only those jobs is returned.

UNLINK *job-number*

unlinks the specified spooler job from the batch job.

[Table 4-3](#) shows the batch job states for which each BATCH subcommand is valid.

Table 4-3. Batch Job States During Which BATCH Subcommands Can Be Used

BATCH Subcommand	Hold (H)	Print (P)	Ready (R)	Open (O)
COPIES	H			
DELETE	H	P	R	
FORM	H			
HOLD	H	P	R	
HOLDAFTER	H	P	R	O
LINK				
LOC	H			
OWNER	H			
REPORT	H			
SELPRI				
START	H			
STATUS				
UNLINK				

Considerations

- To change the attribute values of all jobs belonging to a batch job, you must use the BATCH command.
- You can still use the JOB command to change the attributes of a single job even if that job is part of a batch job; however, you cannot change any of its key attributes (form, owner, or location).

Examples

1. This three-step example requires authorized access. It shows two jobs associated with batch 233: job 7 is the log file, and job 9 has been linked to the log file by the same batch number.

```
) JOB
```

JOB	BATCH	STA	FLAGS	OWNER	TIME	COPY	PAGE	REPORT	LOCATION
1		RDY	4	1,28	11/22	1	7	DEV JAN	#S.PRTR
6		RDY	4	1,44	11/16	1	1	MNL BOB	#SAUR
7	233	RDY	4	1,32	11/16	1	28	ADMN SUE	#LOG
9	233	RDY	4	1,32	11/16	1	100	ADMN SUE	#LOG

```
)
```

2. This command changes the state, location, and priority for all jobs in batch 233:

```
) BATCH 233, HOLD, LOC #LP8, PRI 7, START
```

To verify that the priority for printing batch 233 has changed from 4 to 7, enter a job command:

```
) JOB
```

JOB	BATCH	STA	FLAGS	OWNER	TIME	COPY	PAGE	REPORT	LOCATION
1		RDY	4	1,28	11/22	1	7	DEV JAN	#S.PRTR
6		RDY	4	1,44	11/16	1	1	MNL BOB	#SAUR
7	233	RDY	7	1,32	11/16	1	28	ADMN SUE	#LP8
9	233	RDY	7	1,32	11/16	1	100	ADMN SUE	#LP8

```
)
```

3. The following example shows the difference in the information displayed for a BATCH command with the STATUS option and for a BATCH command with the STATUS and DETAIL options. (The STATUS DETAIL display for job 9 is abbreviated in the example.) For a description of these parameters, see the description of the JOB STATUS display under [JOB Command](#) on page 4-48.

```
) BATCH 233, STATUS
```

JOB	BATCH	STA	FLAGS	OWNER	TIME	COPY	PAGE	REPORT	LOCATION
7	233	RDY	7	1,32	11/16	1	28	ADMN SUE	#LP8
9	233	RDY	7	1,32	11/16	1	100	ADMN SUE	#LP8

```
)
```

```
) BATCH 233, STATUS DETAIL
```

JOB: 7
STATE: READY
LOCATION: #LP8
FORM:
REPORT: ADMN SUE
HOLD BEFORE PRINT: NO
HOLD AFTER PRINT: NO
ABNORMAL: NO
SELECTION PRIORITY: 7
PAGE SIZE: 60
CREATOR ACCESS ID: 1,32
COPIES: 2
PAGES: 28
TOTAL LINES: 1652
OPEN TIME: 16 OCT 88, 13:58:10
CLOSE TIME: 16 OCT 88, 14:23:33
DATA FILE: \$SP.SPLR.SDATA
COLLECTED BY: \$S
UNITS ALLOCATED: 15
BATCH ID: 233
BATCH NAME:
MAXIMUM LINES:
MAXIMUM PAGES:

JOB: 9
STATE: READY
:
:
UNITS ALLOCATED: 15
BATCH ID: 233
BATCH NAME:
MAXIMUM LINES:
MAXIMUM PAGES:

)

COLLECT Command

If you are an authorized user, you can use the COLLECT command to specify attributes, obtain the status, and change the state of the spooler collectors.

Refer to the *Guardian User's Guide* for a description of collector states and default attributes.

Any user can obtain the status of spooler collectors.

```
COLLECT [ $process-name ] [ , subcommand ] ...
```

\$process-name

is the name of the collector to which the subcommands refer. If no collector with this name exists, a new one is created if one or more subcommands are specified. The first character of the process name is always \$ unless it is a remote process where `\system` can precede *\$process-name*.

All the subcommands except STATUS require that a *\$process-name*.

subcommand

is one of the subcommands described below. If *subcommand* is omitted, the STATUS subcommand is assumed.

BACKUP *backup-cpu*

specifies the processor (-1 through 15) that is to run the collector backup. If -1 is specified, the collector does not have a backup. The default specification is -1; that is, the collector has no backup. Your COLLECT command must specify the primary processor number before specifying the backup.

CPU *cpu*

specifies the processor (0 through 15) that is to run the collector. The default processor is the same processor as the supervisor.

DATA *data-file-name*

is the name of the disk file (in the same system) where the collector stores jobs. You must have already created the data file (see [Considerations](#) on page 4-14). This attribute must be specified—there is no default value.

DELETE

removes the specified collector from the spooler subsystem.

DRAIN

causes the collector to stop accepting new jobs. The collector becomes dormant when all open jobs finish spooling.

The collector must be in the dormant state in order to have its attributes changed.

FILE *program-file-name*

specifies the file name of the program to run the collection process. Usually this is `$SYSTEM.SYSTEM.CSPOOL`.

LISTOPENS / **OUT** *file-name* /

provides a list of those jobs that the specified collector currently has open, along with the processor and process identification number (PIN) of the process that spooled the job.

If **OUT** *file-name* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. *file-name* must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

PAGESIZE *num-lines*

specifies a new default job page size. The standard default size is 60 lines per page. This default size can be changed to a new value, ranging from 1 through 127 lines per page. Setting a new default page size is applicable to level-1, level-2, and level-3 spooler applications on a collector basis. If an application specifies a page size for itself, that value will override this default value.

PRI *process-priority*

specifies the execution priority of the collector. The range for this parameter is 1 through 199. The default priority is 145.

START

causes a dormant collector to become active.

STATUS [/ **OUT** *file-name* /] [**DETAIL**]

obtains the status of all collectors in the spooler subsystem.

OUT *file-name*

indicates where Spoolcom is to write the collector status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *file-name* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. *file-name* must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

DETAIL

requests a complete list of all collector attributes. See [Examples for All Users](#) on page 4-16 for a description of the STATUS DETAIL display.

If *\$process-name* is not specified, then the status of all collectors in the spooler subsystem is returned.

If *\$process-name* is specified, then the status of only that collector is returned.

UNIT *unit-size*

is the amount of collection file space allocated to a job initially and each time additional space is needed. It is expressed in units of 1024 bytes (512 words). The valid range is 2 through 32767. Once a collector has been started, the setting of UNIT cannot be altered, even if jobs have been collected. See [Considerations](#) on page 4-14 for more information.

Considerations

- If neither *\$process-name* nor *subcommand* is specified, Spoolcom displays the status of all collectors.
- Any user can get the status of a collector.
- The DETAIL parameter of the STATUS subcommand provides a complete list of collector attributes.
- Only authorized users can modify collector attributes.
- The collector must be in the dormant state before its attributes can be specified or changed.
- The COLLECT command must specify the CPU subcommand before the BACKUP subcommand. If the specified backup processor number is the same as the primary processor number, no backup is created.
- The *data-file-name* parameter should exist before the DATA subcommand is specified. The DATA *data-file-name* subcommand should be specified only before the collector is first started. See the information on creating data files in the *File Utility Program (FUP) Reference Manual*.

- File space usage efficiency is affected by the UNIT size setting if it is significantly larger than the average job size. Since the collector file is not limited by the number of units and UNIT allocation is very efficient, it is recommended that UNIT be set to 4 unless jobs average much less than 4K.
- The collector uses both the collection file buffer size and its UNIT size to establish the size of the buffers it uses in job collection for each opener. File buffer size is established when a file is created (but can be altered later). Up to 4K buffers can be applied to the collector for each opener. The virtual memory used for the buffers is obtained from the Kernel Managed Storage Facility (KMSF).
- The collector establishes the buffer size when it is started. It operates with its collection file only if at least one of the following is true:
 - UNIT size (bytes) is a whole multiple of the buffer size.
 - Buffer size is a whole multiple of the UNIT size (bytes).

If a file that had previously been used by the collector is no longer usable because of a changed buffer size, correction involves establishing a buffer size that obeys at least one of these rules and then restarting the collector. Unless extraordinary file space or virtual memory concerns dominate the decision, it is recommended that the file buffer size be set to 4K and the UNIT setting be a whole multiple of 4.

- You declare a collector by specifying *\$process-name* and initialize it by specifying its other attributes. You can do this with a single COLLECT command, or you can use multiple commands.
- A call to DEVICEINFO or DEVICEINFO2 returns a device subtype even for a process. The subtype returned for the spooler collector is 31.
- The spooler collector does not allow WRITEREADs with read counts greater than zero.
- The job numbers returned by LISTOPENS are not necessarily in sequential order, because they are not stored sequentially in the collector.
- Avoid using the name CSPOOL for any purposes other than as the value for *program-file-name* for the collector.
- When the collector first initializes its collector file, it determines the file's primary, secondary, and maxextent sizes and expands the file to its maximum possible size during initialization. It will then create as many units as will fit into this expanded size and write an end-of-file (EOF) mark at the end of the maximum size.

COLLECT STATUS Display

The STATUS subcommand can be issued at any time. It produces a display with the following headings:

```
COLLECT STATE  FLAGS CPU  PRI  UNIT  DATA FILE  %FULL
```

[Table 4-4](#) describes the COLLECT STATUS display entries.

Table 4-4. COLLECT STATUS Display Entries

Entry	Description								
COLLECT	Is the collector name.								
STATE	Is the state of the collector. STATE can be one the following: <table border="0" style="margin-left: 20px;"> <tr> <td>ACTIVE</td> <td>The collector process is running. This means that it is either collecting data or waiting to collect data.</td> </tr> <tr> <td>DORMANT</td> <td>The collector process is not running.</td> </tr> <tr> <td>DRAIN</td> <td>The collector process has stopped accepting new jobs and will enter the dormant state when all open jobs have finished spooling.</td> </tr> <tr> <td>ERROR %nnn</td> <td>The collector process has failed. It must be removed from this state before it can become active again. When the collector is in the error state, the octal number that follows ERROR indicates the error condition. It can be either %1000 plus a file-system error number or %100000 plus a NEWPROCESS error number.</td> </tr> </table>	ACTIVE	The collector process is running. This means that it is either collecting data or waiting to collect data.	DORMANT	The collector process is not running.	DRAIN	The collector process has stopped accepting new jobs and will enter the dormant state when all open jobs have finished spooling.	ERROR %nnn	The collector process has failed. It must be removed from this state before it can become active again. When the collector is in the error state, the octal number that follows ERROR indicates the error condition. It can be either %1000 plus a file-system error number or %100000 plus a NEWPROCESS error number.
ACTIVE	The collector process is running. This means that it is either collecting data or waiting to collect data.								
DORMANT	The collector process is not running.								
DRAIN	The collector process has stopped accepting new jobs and will enter the dormant state when all open jobs have finished spooling.								
ERROR %nnn	The collector process has failed. It must be removed from this state before it can become active again. When the collector is in the error state, the octal number that follows ERROR indicates the error condition. It can be either %1000 plus a file-system error number or %100000 plus a NEWPROCESS error number.								
FLAGS	Shows a D if the collector is in debug mode.								
CPU	Gives the numbers of the processors running the primary and backup processes when the collector was started.								
PRI	Is the execution priority of the collector.								

Examples for All Users

- To show the status of all collectors on the system, enter

```
) COLLECT, STATUS
```

The following information is displayed:

```
COLLECT STATE  FLAGS CPU  PRI  UNIT  DATA FILE  %FULL
$C      ACTIVE      3 , 10 146 16  $ACC.SPOOLER.DATA 0
$G      ACTIVE      4 , 5  146 8  $PUB.SPOOLER.DATA 5
)
```


2. To show the status of only one collector, enter

```
) COLLECT $C, STATUS
```

The following information is displayed:

```
COLLECT STATE FLAGS CPU    PRI UNIT DATA FILE          %FULL
$C      ACTIVE      3 ,10 146 16  $ACC.SPOOLER.DATA 0
)
```

3. To find more information about that one collector, specify STATUS DETAIL:

```
) COLLECT $C, STATUS DETAIL
```

The following information is displayed:

```
COLLECTOR: $C
STATE: ACTIVE
LAST ERROR: NONE
PROGRAM FILE: $SYSTEM.SYSTEM.CSPOOL
CPU: 3
BACKUP: 10
PRIORITY: 146
DATA FILE: $ACC.SPOOLER.DATA
PAGESIZE: 60
UNIT SIZE: 16
ALLOCATED UNITS: 1
TOTAL UNITS: 200
PERCENT FULL: 0
)
```

4. To list the jobs that the specified collector currently has open, specify

```
) COLLECT $S1, LISTOPENS
```

If the command is entered and no jobs are open, the following message is displayed:

```
THERE ARE NO JOBS CURRENTLY OPEN
)
```

When there are open jobs, the data is displayed in the following format:

```
JOB      PROCESS      CPUPIN
321      $TPMS        09,057
2320     $ZCD3        10,003
)
```

Examples for Authorized Users

1. The following command specifies that CPU 3 is the primary location (and CPU 0 is the backup) from which collector \$S is to run a copy of the program located in \$SYSTEM.SYSTEM.CSPOOL. This collector stores jobs on the file SPLDATA:

```
)COLLECT $S, FILE CSPOOL, DATA SPLDATA, CPU 3, BACKUP 0
```

2. The following command stops the collector from accepting any new jobs, causing it to enter the dormant state when all jobs have finished spooling:

```
)COLLECT $S, DRAIN
```

COMMENT Command

The COMMENT command is used to insert explanatory material, which is ignored by Spoolcom, into a Spoolcom command file. It is often used in command files (sometimes called OBEY files) to explain the purpose of the file and what the commands in it do. Command files are often used as an easy-to-execute and uniform procedure for starting the spooler.

```
COMMENT [ any-text ]
```

any-text

is text that is ignored by Spoolcom. *any-text* is terminated by a carriage return or a semicolon (;).

Consideration

- At least one blank space must separate the keyword COMMENT from the text.

Examples

See the command files listed in the *Guardian User's Guide*. These files allow you to coldstart and warmstart the spooler.

DEV Command

The DEV command controls and obtains the status of devices in the spooler subsystem.

Authorized users can use the DEV command to delete a device from the spooler or to add to the spooler a new device that was configured at system generation but was not included in the spooler coldstart.

Refer to the *Guardian User's Guide* for a description of these tasks as well as information on device states and default attributes.

All users can obtain the status of any device as well as perform some of the control subcommands.

```
DEV [ $device ] [ , subcommand ] ...
```

\$device

names the device to which the subcommands refer. All the subcommands except STATUS and XREF require that this parameter be included. The first character of the device name is always \$ unless it is a remote process where `\system` can precede *\$device*.

subcommand

is one of the subcommands described below. If *subcommand* is not specified, STATUS is assumed. See [Table 4-6](#) for a list that shows which subcommands are valid in which DEV states.

ALIGN

prints on the specified device a page containing the device form alignment template (a measured ruler line). This enables you to see how many characters to a line the device can print, as well as to verify the paper alignment and print quality. ALIGN assumes HEADER OFF and TRUNC ON. The align template is truncated to the *device-width* setting if WIDTH is set to less than 132. A header is never printed with the align template.

The align template appears as JOB 0 in the device queue, scheduled as the next job to print. The align job will print even when the device is offline. If another job is currently printing, the align template will print after the current job finishes.

CHARMAP [NONE | IBMKANJIKANA | JEFKANJIKANA | JISKANJIKANA]

indicates whether a multibyte character-set translation is required and, if so, what type of printer is being used.

`NONE` specifies that no multibyte character-set translation is required. This is the default value if the `CHARMAP` subcommand is not used or is used without a parameter.

`IBMKANJIKANA` specifies the IBM Kanji character set.

`JEFKANJIKANA` specifies the Fujitsu Kanji character set.

`JISKANJIKANA` specifies the JIS C-6234 or C-6220 character set.

`CLEAR [DEL]`

stops printing the current job. If `DEL` is specified, the job is deleted from the spooler subsystem; otherwise, the `DEV` command returns the job to the end of the device queue. The device remains online so the next job in the device print queue can begin printing.

`DELETE`

removes the device from the spooler subsystem. This subcommand works only if the device is not connected to any location (that is, it has been disconnected by the `LOC DEV` command or removed with the `LOC DELETE` command) and has no jobs in its queue (that is, it has been drained using the `DEV DRAIN` command).

`DEVRESET [ON | OFF]`

specifies whether the spooler print process resets both the print device and the print input/output process (IOP) at the beginning of each print job. This subcommand is supported only by `FASTP` and the Universal Interface (UI) IOP (parallel print process).

If `DEVRESET` is not specified, it defaults to `OFF`. If `DEVRESET` or `DEVRESET ON` is specified, it causes the spooler print process to reset the print IOP and the print device to its configured default values at the beginning of each print job.

`DEVRESET OFF` causes the spooler print process to reset the print IOP, but not the print device, to its configured default values at the beginning of each print job. See [Considerations](#) on page 4-33 for exceptions.

`DEVTYPE [blank | LU1 | LU3 | 7 | 8 | 9 | 10 | 5512 | 5515 | 5516 | 5518 | 5573 | 5574 | 5577]`

The default value is `blank`.

This parameter is required only when the IOP to which the device is attached cannot determine the device type (a `DEVICE INFO` command does not return a device type). A connection to a LAN printer is an example of a situation where the `DEVTYPE` parameter is required.

`DEVTYPE blank` sets the print device configuration to `blank`.

DEVTYPE LU1 and DEVTYPE LU3 control the configuration of a print device attached to a controller located in an IBM domain. FASTP must be the spooler print process. DEV PARM form-feed options take precedence over all DEVTYPE LU1 and LU3 subcommands. If either FASTP is not the current spooler print process, Spoolcom accepts the DEVTYPE LU1 or LU3 subcommand but the print process ignores it.

DEVTYPE LU1 configures the print device for SNA character string (SCS) mode.

DEVTYPE LU3 configures the print device for data stream compatibility (DSC) mode.

DEVTYPE 7, 8, 9, or 10 controls the configuration of a print device for specific types of printer, identified by the device subtype:

- DEVTYPE 7 configures the print device for the 5515, 5516, and 5518 printers.
- DEVTYPE 8 configures the print device for the 5573 and 5574 printers.
- DEVTYPE 9 configures the print device for the 5512 printer.
- DEVTYPE 10 configures the print device for the 5577 printer.

DEVTYPE can also be used to control the configuration of a print device by the printer product number:

- DEVTYPE 5512 configures the print device as the 5512 printer.
- DEVTYPE 5515 configures the print device as the 5515 printer.
- DEVTYPE 5516 configures the print device as the 5516 printer.
- DEVTYPE 5518 configures the print device as the 5518 printer.
- DEVTYPE 5573 configures the print device as the 5573 printer.
- DEVTYPE 5574 configures the print device as the 5574 printer.
- DEVTYPE 5577 configures the print device as the 5577 printer.

\$device must be in the offline state (DEV *\$device*, DRAIN) before the DEVTYPE subcommand is issued.

DRAIN

causes the device to go offline after the job currently printing is finished. If the device is in an invalid state, Spoolcom resets the invalid state before taking the device offline.

ENDFF [ON | OFF]

specifies whether the spooler print process, FASTP, issues a form feed at the end of each job.

ENDFF or ENDFE ON specifies that the print process will issue a form feed or other command to the printer to advance to a new page at the end of each job. The command varies depending on the printer type. For printers that can recognize the conditional operation, this advance is conditional so that extra blank pages do not occur if the printer is already at top-of-form when the job ends. For cut-sheet printers such as the 5573, 5574, and 5575, this advance ejects the last printed page.

ENDFF OFF specifies that the print process will not issue a form feed at the end of each job. This is the default device configuration value if you do not specify the ENDFE subcommand.

See [Considerations](#) on page 4-33 for more information about using ENDFE with FASTP.

EXCLUSIVE [ON | OFF [!]]

specifies the ownership mode of the device.

EXCLUSIVE or EXCLUSIVE ON specifies that the print process should not close the device when the printer is not printing a job.

EXCLUSIVE OFF (the default value) specifies that the print process should close the device when no more jobs are waiting to be printed.

EXCLUSIVE OFF ! specifies that the device is to be closed between jobs.

FIFO [ON | OFF]

specifies the algorithm by which jobs are selected for printing on the device. If FIFO or FIFO ON is specified, first-in, first-out (FIFO) queuing is selected.

FIFO OFF (the default algorithm) specifies that jobs are not queued first in, first out, but rather according to an algorithm that takes into account the relative length of an incoming job and how long the other jobs have been in the queue.

A description of the queuing algorithm appears in the *Guardian User's Guide*.

FORM [*form-name*]

specifies a form name for the device. The name is composed of as many as 16 letters, numbers, or blanks. If FORM is present but *form-name* is omitted, a blank form name is assigned.

The default name is a blank form name.

HEADER [ON | OFF | BATCH]

specifies whether a standard or batch header page should precede each job. The ON, OFF, and BATCH options are supported by the FASTP print processes.

HEADER or HEADER ON (the default specification) specifies that a header page should print at the beginning of every job.

HEADER OFF specifies that header pages are not to be printed at the beginning of jobs.

HEADER BATCH specifies that two header pages and three trailer pages (also containing job information) print for every job. This enables jobs printed on accordion-fold paper to have a header page appearing on top regardless of whether the job begins on an odd or even page. Printing over the page folds on the trailer pages enables jobs to be easily separated from each other.

HEADER ON and HEADER BATCH assume TRUNC ON.

JOB job-number

causes the specified job to be placed up in the print queue to be the next job printed. By placing the device in the offline state, you can use this subcommand to manually select a job that you want printed.

This subcommand does not move an entry already in a print queue. Instead, it makes a new entry that is placed at the head of the queue. Thus, any preexisting queue entries for the same job remain. If the device is in the offline state, this job (and only this job) will print and the device will return to the offline state. If the device is currently printing another job, this job will be placed at the front of the device queue and will be the next job to print. If *job-number* is currently printing, no action is taken and no error is returned.

If the device is offline and you submit more than one of these job subcommands (specifying different jobs), only the first job will print. The device will return to an offline state after the first job prints, preventing the others from printing. All jobs specified with this subcommand are copied to the head of the queue, even if they cannot print.

Jobs selected to print in this manner will have a higher priority than other jobs regardless of the scheduling algorithm (FIFO ON/OFF) or the priority of other jobs. See Example 4 of [Examples for Authorized Users](#) on page 4-38.

LUEOLVALUE [NL | CRLF]

sets the end-of-line (EOL) sequence that the FASTP print process places at the end of a print record for LU1 and LU3 type printers only. Other printer types are not affected by this subcommand. This subcommand applies only to the FASTP print process.

If CRLF is specified, the default EOL sequence is carriage return and line feed as follows: CR (%15) and LF (%12).

If new line (NL) is specified, the end-of-line sequence is NL (%205, IAW 32876-2).

LUEOLWHEN [LT132 | LTWIDTH | ALWAYS | NEVER]

indicates when the FASTP print process is to place an EOL sequence at the end of a print record for LU1 and LU3 type printers (other printer types are not affected). This subcommand applies only to the FASTP print process.

If no option is specified or if LT132 is specified, FASTP places the EOL sequence at the end of the print record if the byte length is less than 132. This is the default placement. If the byte length is 132 or greater, the controller adds the EOL sequence.

If LTWIDTH is specified, FASTP places the EOL sequence at the end of the print record if the byte length is less than or equal to the device print width (set by the Spoolcom command DEV WIDTH).

If ALWAYS is specified, FASTP always places the EOL sequence at the end of the print record.

If NEVER is specified, FASTP never places the EOL sequence at the end of the print record. The application must assume the responsibility of placing the EOL sequence.

LUTOFVALUE [CRFFCR | FFCR | FF | NEVER]

sets the sequence that designates top of form (new page) that the FASTP print process is to use for a print record for LU1 and LU3 type printers (other printer types are not affected). This subcommand applies only to the FASTP print process.

If CRFFCR is specified, the default top-of-form sequence is carriage return, form feed, and carriage return as follows: CR (%15), FF (%14), and CR (%15).

If FFCR is specified, the top-of-form sequence is FF (%14) and CR (%15).

If FF is specified, the top-of-form sequence is FF (%14).

If NEVER is specified, no top-of-form sequence is written. The application must assume the responsibility of placing the top-of-form sequence.

PARM *parameter*

specifies a device parameter that is passed to the print process controlling this device whenever the supervisor communicates with the print process. The range for *parameter* is -32768 through +32767. The default is 0.

- The FASTP print process uses *parameter* as follows:
 - <0 : 8> Reserved. Set to 0.
 - <9> = 0 Print headers in the normal orientation, which is portrait. For 5577 printers, you should disable offset of jobs on output.
 - <9> = 1 For jobs sent to 5573, 5574, and 5577 printers, set 10-point, 12-cpi landscape text for header and trailer pages. For 5577 printers, issue a PCL job-separation command before printing the job (which causes each job to be offset in the output bin).
 - <10> = 0 For jobs transferred to a target spooler collector, retain the same job ownership at the target spooler as at the source spooler.
 - <10> = 1 For jobs transferred to a target spooler collector, set job ownership at the target node to the print process (PROCESSACCESSID) performing the transfer.
 - <11> = 0 For jobs transferred to a target spooler collector, set the HOLDAFTER flag as indicated in bit <12>. This bit is ignored for jobs sent to devices other than spooler collectors.
 - <11> = 1 For jobs transferred to a target spooler collector, always set the HOLDAFTER ON flag. This bit is ignored for jobs sent to devices other than spooler collectors.
 - <12> = 0 For jobs transferred to a target spooler collector, always set HOLDAFTER OFF. This bit is used only if bit <11> = 0. This bit is ignored for jobs sent to devices other than spooler collectors.
 - <12> = 1 For jobs transferred to a target spooler collector, always retain any HOLDAFTER value specified for the job at the source spooler when transferred to the target spooler. This bit is used only if bit <11> = 0. This bit is ignored for jobs sent to devices other than spooler collectors.
 - <13> = 0 Set target location name to that of the source job location.
 - <13> = 1 Set target location name to that of the device thread name.

- <14> = 0 Set paper length to 66 lines (the default).
- <14> = 1 Set paper length to 72 lines rather than the default 66 lines. This information applies for printing trailer pages (HEADER BATCH).
- <15> = 0 Print any header and trailer outside the scope of any font job.
- <15> = 1 Print any header and trailer pages within the scope of any font job. For example, if a font job selects landscape printing on a 5573 printer, setting this bit sets landscape for the header and trailer rather than the default (portrait) mode.

PREXLATE [OFF | ON]

specifies postspacing (vertical space after printing a line) as the default vertical spacing mode. This subcommand is optional; the default specification is OFF, indicating that no prespace-to-postspace translations are performed. This subcommand applies only to the FASTP print processes and to printers connected through the Universal Interface controller (subtypes 1, 5, and 7).

PROCESS *\$process-name*

specifies the process name for the print process that controls the device. This subcommand is required; there is no default name. The first character of the process name is always \$.

RESTART [OFF | *interval* | ON]

specifies whether or not a supervisor process should attempt to start a device after the device has been placed offline because of a nonretryable I/O error. The default value is RESTART OFF.

- RESTART OFF: No attempt is made to restart the device.
- RESTART or RESTART ON: An attempt is made to restart the device after an interval of 120 seconds.
- RESTART *interval*: an attempt is made to restart the device after a value of *interval* seconds. The interval range is 10 through 32767 seconds.

If RESTART, RESTART ON, or RESTART *interval* are specified, a supervisor process attempts to restart the device indefinitely.

Use RESTART to specify the action to be taken for a nonretryable I/O error. Use RETRY and TIMEOUT to specify the action to be taken for a retryable I/O error.

RETRY *interval*

specifies the number of seconds a print process waits before retrying an I/O operation that has failed because of a retryable I/O error. The valid range for *interval* is 1 through 32767. The default value is 5 seconds.

Table 4-5. Retryable Error Numbers

Retryable Error Number	Description
12	file in use
100	device not ready
102	paper out
140	modem error
218	interrupt timeout

Use RESTART to specify the action to be taken for a nonretryable I/O error. Use RETRY and TIMEOUT to specify the action to be taken for a retryable I/O error.

SKIP [+ | -] *number-of-pages*

causes the device to skip ahead or skip back *number-of-pages* relative to the current page. A minus sign (-) indicates skipping backward; no sign or a plus sign (+) indicates skipping forward. The device starts printing at the first line on the new page. The range of values for *number-of-pages* is -32767 through +32767. Users who are not a system operator (user ID 255, *n*) can execute this subcommand only on their own jobs.

SKIPTO *page-number*

causes the device to skip to the indicated page. The range of values for *page-number* is 1 through 65500 pages. Users who are not system operators (user ID 255, *n*) can execute this subcommand only on their own jobs.

SPEED *lines-per-minute*

allows you to specify the estimated printing speed of the device. This subcommand is used for calculating how long a job takes to print. It has no effect on the actual speed of the device. The range of values for *lines-per-minute* is 1 through 32767. If not specified, a default speed of 100 is assumed.

START

causes an offline device to become online. All users can restart devices that have gone offline due to a device error (for example, running out of paper). This subcommand causes a suspended device to resume printing.

Devices taken offline can be restarted only by an authorized user.

STARTFF [ON | OFF [!]]

specifies whether the spooler print process FASTP issues a form feed at the beginning of each job.

If STARTFF or STARTFF ON is specified, FASTP issues a form feed at the beginning of each job. This form feed is suppressed if the paper is at top-of-form or if the job started with a form feed, thereby preventing extra blank pages.

STARTFF OFF (the default value) specifies that FASTP is not to issue a form feed at the beginning of each job. In this case, FASTP issues a form feed at the start of a job if HEADER ON or HEADER BATCH is configured. For jobs that do not contain a starting form feed, this default action can result in jobs printed without an inserted blank page between jobs. Inserting blank pages can be altered by the ENDDFF, DEVRESET, or EXCLUSIVE subcommands. If STARTFF is not specified, it defaults to OFF.

STARTFF OFF ! specifies that FASTP ignore the first form feed in the print job based on the assumption that there are no paper motion commands before the form feed.

Note that combinations of conditional form feeds as a result of STARTFF ON, ENDDFF ON, DEVRESET ON, and EXCLUSIVE OFF [!] will not result in multiple form feeds by the device.

STATUS [/ OUT *filename* /] [DETAIL]

displays the attributes of devices in the spooler subsystem. See [DEV STATUS Display](#) on page 4-35 for an example of this display.

OUT *filename*

indicates where Spoolcom is to write the device status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

DETAIL

requests a complete list of all of the device attributes. If $\$device$ is not specified, then the status of all devices in the spooler subsystem is returned.

SUSPEND

causes the device to suspend printing of the job currently printing. The same job resumes printing when the device is restarted with the START subcommand.

TIMEOUT *number-of-retries*

specifies the number of times a print process retries a retryable I/O operation that has failed. The value for *number-of-retries* can be either in the range 1 through 32767, or equal to -1 (-1 means the print process retries a retryable operation indefinitely). The default value is 360. See [Table 4-5](#) for a list of the retryable error numbers.

Use RESTART to specify the action to be taken for a nonretryable IO error. Use RETRY and TIMEOUT to specify the action to be taken for a retryable IO error.

An additional timeout device attribute is available that specifies the number of seconds a FASTP print process waits before considering an I/O operation to be a failure.

Instead of being configured with the Spoolcom DEV command, this device attribute is read from a FASTCNFG print process configuration file. Refer to the *LAN Print Spooler Manual* for information about:

- Configuring a FASTCNFG file
- Setting the Spoolcom DEV PARM value to make a FASTP print process read the FASTCNFG file

The timeout device attribute is specified by the keyword TIMEOUT followed by an equal sign (=) and a numeric value. The numeric value is the number of seconds FASTP will wait before considering an I/O operation to the device a failure. If the TIMEOUT attribute is omitted or if it has a value of 0, FASTP waits indefinitely or until the I/O process for the device declares the operation a failure.

The following FASTCNG keywords and keyword attributes are required when you specify the TIMEOUT attribute:

```
DEVICE $spooler-device-name
CONTYPE = NONE
```

Examples of FASTCNFG file entries:

```
/* timeout = 120 seconds. */
DEVICE $DEV1
CONTYPE = NONE
TIMEOUT = 120

/* timeout = 0 seconds. */
DEVICE $Dev2
```

```

CONTYPE = None
TIMEOUT = 0

/* timeout = 0 seconds. */
DEVICE $dev3
CONTYPE = none

```

TRUNC [ON | OFF]

specifies whether lines longer than the device width are to be truncated or wrapped around. TRUNC or TRUNC ON causes the print process to discard the ends of spooled lines that are longer than the device width. TRUNC OFF (the default) causes the print process to print on the next sequential line the ends of any spooled lines that are longer than the device width.

TRUNC ON is always used when printing headers.

WIDTH *device-width*

specifies the maximum line length for the device. The range of values for *device-width* is 0 through 32676. A value of -1 (the default value) causes the print process for the device to obtain the record size from the file-system DEVICEINFO procedure. A value of 0 indicates there is no limit to width.

XREF [/ OUT *filename* /]

produces a cross-reference list of devices, locations, and print processes, ordered by device. If *\$device* is not specified, a complete cross-reference of all devices is produced.

If OUT *filename* is not specified, the list is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, an unstructured file, a relative file, an entry-sequenced file, a spooler job file, a process, or a device.

[Table 4-6](#) shows the device states for which each DEV subcommand is valid.

Table 4-6. Device States During Which DEV Subcommands Can Be Used (page 1 of 2)

DEV Subcommand	Offline (O)	Busy (B)	Waiting (W)	Suspended (S)	Deverror (D)	Procerror (P)
ALIGN	O	B	W	S	D	P
CHARMAP	O					
CLEAR		B		S	D	
DELETE	O					
DEVRESET	O					

Table 4-6. Device States During Which DEV Subcommands Can Be Used (page 2 of 2)

DEV Subcommand	Offline (O)	Busy (B)	Waiting (W)	Suspended (S)	Deverror (D)	Procerror (P)
DEVTYPE	O					
DRAIN	O	B	W	S	D	P
ENDFF	O					
EXCLUSIVE	O					
FIFO	O					
FORM	O					
HEADER	O					
JOB	O	B	W	S	D	P
LUEOLVALUE	O					
LUEOLWHEN	O					
LUTOFVALUE	O					
PARM	O					
PREXLATE	O					
PROCESS	O					P
RESTART	O					
RETRY	O					
SKIP		B		S	D	
SKIPTO		B		S	D	
SPEED	O					
START	O			S		
STARTFF	O					
STATUS	O	B	W	S	D	P
SUSPEND		B		S	D	
TIMEOUT	O					
TRUNC	O					
WIDTH	O					
XREF	O	B	W	S	D	P

Considerations

- The device name uses an internal filename format. The device name can only be up to 32 bytes. Therefore, any device with an eight character remote volume name cannot be configured for use with DEV command.
- If neither *\$device* nor *subcommand* is specified, Spoolcom displays the status of all devices.
- Any user can obtain the status of a device.
- The SKIP and SKIPTO subcommands can be executed by the owner of the job currently printing on the device.
- The START subcommand can be executed by a user only if the device is offline and has a device error against it. Unless these two conditions are both true, the standard user cannot use this subcommand.
- Only an authorized user can set and modify the attributes of a device.
- Before you can delete a device, the device queue must be empty (issue a Spoolcom DEV, DRAIN command). Either disconnect the device from its location (with the Spoolcom LOC, DEV command) or delete the location (with the Spoolcom LOC, DELETE command). Any jobs using this location must be deleted before you can delete the location.
- The print device is reset to its configured values each time the spooler print process opens it, regardless of whether you have specified DEVRESET ON. Normally, the spooler print process closes the print device after printing all the jobs on the device queue and opens it again the next time it has jobs to print. Refer to the EXCLUSIVE subcommand for information on controlling when the spooler print process closes the print device.
- If you specify HEADER ON, then STARTFF ON and STARTFF OFF have no effect.
- For the FASTP print process, the SKIP and SKIPTO subcommands can have a delayed effect with certain printers, because several lines or pages might be in transmission to the printer before the SKIP or SKIPTO takes effect.
- The STARTFF subcommand suppresses inserting a blank page between jobs unless HEADER ON or HEADER BATCH is configured. Other subcommands affect inserting a blank page. To ensure that FASTP does not issue form feeds between jobs, so that you can print multiple jobs per page, you should have these subcommands configured as follows:

```
STARTFF OFF  
ENDFF OFF  
DEVRESET OFF  
EXCLUSIVE ON
```
- If you are configuring a laser printer run by the FASTP print process, set ENDFF ON to ensure that a form feed is issued at the end of a print job.

- Do not configure a printer with DEV EXCLUSIVE OFF! if it is associated with locations that are configured with font jobs to affect the printing of a subsequent print job. The EXCLUSIVE OFF! attribute causes the device to be closed and opened between printing the font job and printing the subsequent print job, thereby terminating the effects of the font job. Configure a device in this situation with either EXCLUSIVE ON or EXCLUSIVE OFF.
-
- The DEV subcommands supported by FASTP—the Tandem print process—are as follows:

ALIGN	LUTOFVALUE
CHARMAP	PARM
CLEAR	PREXLATE
DEVRESET	SKIP
DEVTYPE	SKIPTO
ENDFF	STARTFF
HEADER	TRUNC
LUEOLVALUE	WIDTH
LUEOLWHEN	

Note. The following DEV subcommands are not listed above because they either support device status or provide attributes that have no dependencies on an associated print process: DELETE, DRAIN, EXCLUSIVE, FIFO, FORM, JOB, PROCESS, RESTART, RETRY, SPEED, START, SUSPEND, TIMEOUT, and XREF.

Table 4-7. DEV STATUS Display Entries (page 2 of 2)

Entry	Description
D	Draining
T	Truncate on
!	Exclusive off
PROC	is the name of the device print process.
FORM	is the device form name.

Examples for All Users

1. You can obtain the status of one device on the system by entering

```
)DEV $DEV1
```

The following information is returned:

```

DEVICE          STATE          FLAGS  PROC  FORM
$DEV1          JOB 2482      H      $SPLA

  JOB  LOCATION      DEVICE          SEQ  COPY  PAGE
  2482 #LOC1.DEFAULT  $DEV1          1    1    3

  2486 #LOC1.DEFAULT  $DEV1          2    1    3

  2490 #LOC1.DEFAULT  $DEV1          3    1    3
)

```

2. The detailed status of that same device is obtained as follows:

```
)DEV $DEV1, STATUS DETAIL
```

The following information is returned:

```
DEVICE: $DEV1
```

```

STATE: OFF LINE
LAST ERROR: %004016
EXCLUSIVE: OFF
FIFO: OFF
HEADER: ON
TRUNCATION: OFF
DRAINING: NO
PRINTING JOB: NONE
PARM: 0
PROCESS: $PR10
RETRY: 5
TIMEOUT: 360
SPEED: 100
WIDTH: 132
FORM:
RESTART: OFF
DEVRESET: OFF
DEVTYPE:
STARTFF: OFF
ENDDFF: OFF
CHARMAP: NONE
PREXLATE: OFF

LUTOFVALUE: CRFFCR
LUEOLVALUE: CRLF
LUEOLWHEN: LT132

```

)

LAST ERROR is the octal number of the last error printed on the error log file. The %004016 error shown here is translated by reading bits 8 through 15. Bits 8 through 15 contain 00 001 110, which translates to 016 octal or 14 decimal: an error that is typically caused by a device not existing in the system.

3. If the job currently printing is yours, you can use the SKIPTO subcommand to skip ahead in your spooled program to a page number (found by using Peruse).

```
)DEV $DUZ1, SKIPTO 43
```

4. The SKIP subcommand enables you to skip ahead or back a specified number of pages in your program printing. For instance, if the ribbon turned or the paper became mangled, this command allows you to make the printer go back and reprint the last eight pages to produce a clean copy:

```
)DEV $LP3, SKIP -8
```

5. If \$HT1 has gone offline because it ran out of paper, you can restart it (after putting in new paper) with this command:

```
)DEV $HT1, START
```

Examples for Authorized Users

1. In the example below, the print process that is to control device \$LP3 is specified and causes all jobs printed on that device to have a header page:

```
)DEV $LP3, PROCESS $XP, HEADER ON
```

2. The START subcommand is used to bring back online a device that has been taken offline for any reason:

```
)DEV $LP3, START
```

3. The printer \$LP1 is deleted from a running spooler subsystem by first draining the device and then disconnecting it from its location:

```
)DEV $LP1, DRAIN
)LOC #LP1.LP1, DEV
)DEV $LP1, DELETE
```

4. In this example, device \$LP3 is currently printing job 10; and jobs 20, 30, 40, and 50 are in the device queue.

```
)DEV $LP3
DEVICE          STATE          FLAGS  PROC          FORM«$LP3          JOB 10
H              $SPLP

JOB  LOCATION          DEVICE          SEQ  COPY  PAGE
10  #LP3.DEFAULT        $LP3          PRINT 1    10
20  #LP3.DEFAULT        $LP3          2     1    10
30  #LP3.DEFAULT        $LP3          3     1    10
40  #LP3.DEFAULT        $LP3          4     1    10
50  #LP3.DEFAULT        $LP3          5     1    10
```

To make job 40 the next job to print, you can enter the following:

```
)DEV $LP3, JOB 40; DEV $LP3
DEVICE          STATE          FLAGS  PROC          FORM
$LP3            JOB 10          H      $SPLP

JOB  LOCATION          DEVICE          SEQ  COPY  PAGE
10  #LP3.DEFAULT        $LP3          PRINT 1    10
40  #LP3.DEFAULT        $LP3          2     1    10
20  #LP3.DEFAULT        $LP3          3     1    10
30  #LP3.DEFAULT        $LP3          4     1    10
50  #LP3.DEFAULT        $LP3          5     1    10
```

5. A typical configuration for a laser printer, type 5577, to use the FASTP print process is shown in [Table 4-8](#) as reported by a detailed DEV STATUS command. The DEV commands to configure this printer are shown following the status listing.

Table 4-8. Typical Configuration for Laser Printer Type 5577

Device Attribute	Default?	Description
EXCLUSIVE: OFF	Yes	Close the device when no jobs are waiting.
FIFO: OFF	Yes	Use the queuing algorithm that considers the relative job length and the time jobs have been queued.
HEADER: ON	Yes	Print the banner at the beginning of every job.
TRUNCATION: OFF	Yes	Print wrapped text on the next line.
PARM: 64 (%100)	No	Use landscape headers.
PROCESS: \$SPLX	No	Use the FASTP print process. \$SPLX is suggested for the FASTP process name. There is no default value for the PROCESS subcommand, but the specified name must have been associated with FASTP through a previously issued PRINT command.
RETRY: 5	Yes	Wait 5 seconds before retrying a failed retryable operation to the device.
TIMEOUT: 360	Yes	Retry a failed retryable operation to the device 360 times.
SPEED: 900	No	Use this speed for estimating print time.
WIDTH: -1	Yes	Use the record size obtained by the file-system DEVICEINFO procedure for the maximum line length.
RESTART: 300	No	Wait 300 seconds before a restart attempt.
DEVRESET: ON	No	Reconfigure the print IOP and the print device to their default values at the beginning of each print job.
DEVTYPE: 10	No	Configure the device for a 5577 printer.
STARTFF: OFF	Yes	Suppress a form feed at beginning of job.
ENDFF: ON	No	Eject page at end of job.
CHARMAP: NONE	Yes	No multibyte character set is used.
PREXLATE: OFF	Yes	Perform no prespace-to-postspace translations.
LUTOFVALUE: CRFFCR	Yes	N/A
LUEOLVALUE: CRLF	Yes	N/A
LUEOLWHEN: LT132	Yes	N/A

The following DEV commands implement the configuration shown in [Table 4-8](#):

```
DEV $LZ5577.#A, DEVTYPE 10, PARM 64, PROCESS $SPLX
DEV $LZ5577.#A, SPEED 900, RESTART 300, DEVRESET ON, ENDDFF ON
```

EXIT Command

The EXIT command terminates an interactive Spoolcom session.

```
EXIT
```

Example

1. After executing Spoolcom commands, use EXIT to return to the command interpreter:

```
)JOB 1855, HOLD, LOC #HT1, START  
)EXIT
```

```
2>
```


FC Command

FC, or Fix Command, allows you to modify and resubmit the last command line entered. The FC edit subcommands are similar to those used for the TACL FC command, described in the *TACL Reference Manual*.

FC

Consideration

- The FC command executed in Spoolcom can modify only the last line, unlike an FC command executed at a TACL prompt, which can identify and modify other lines.

Example

In the following example, the FC command corrects a syntax error:

```
)DEV #LP, STATUS
```

```
^
```

```
INVALID COMMAND PARAMETER
```

```
)FC
```

```
)DEV #LP, STATUS
```

```
. $(press RETURN)
```

```
)DEV $LP, STATUS
```

```
.(press RETURN)
```

DEVICE	STATE	FLAGS	PROC
FORM			
\$LP	WAITING	T	\$SPLP

FONT Command

The FONT command causes the spooler to recognize a job as a font job and to associate that job with a specified font name. A font job contains commands that can be sent to a programmable printer to control such features as character sets, vertical forms-control specifications, or compressed printing. Once a font job is defined, you can use the LOC FONT subcommand to associate it with a spooler location.

If you are an authorized user, you can use the FONT command to define a font job, delete a font job associated with a specific font name, or obtain the status of a job that is associated with a specific font name.

Refer to the *Guardian User's Guide* for more information on configuring a printer with special-control formatting capabilities using the FONT command.

All users can display the status of all font jobs or of a job that is associated with a specific font name.

```
FONT [ font-name [ , subcommand ] ]
```

font-name

identifies the specific font name associated with the job number. The *font-name* parameter can be specified as 1 through 16 letters or digits; the first character must be a letter.

subcommand

is one of the subcommands described below. If *subcommand* is not specified, STATUS is assumed. Users who are not system operators (user ID 255, *n*) can use these subcommands to change the attributes and status of their own jobs only.

DELETE

specifies a font name to be deleted.

```
JOB [ job-number | ( qualifiers ) ]
```

requires either a job number or sufficient qualifiers to uniquely identify a job.

If neither *job-number* nor *qualifiers* is specified, the spooler specifies the same font name for all jobs on the system.

job-number

specifies a particular job to be associated with *font-name*.

(*qualifiers*)

specify the conditions that a job must meet in order to be associated with the font name. The *qualifiers* parameter is any number or combination of the following qualifiers, separated by commas and all enclosed in parentheses. If more than one job meets the specified qualifications, then only the first to do so is used. Each qualifier can be specified only once.

COLLECT *\$process-name*

selects all jobs collected by this collector.

DATE { FROM *time* [THRU *time*] | THRU *time* }

selects all the jobs created at the specified date and time or within the specified range of dates and times. If DATE is omitted, today's date is used; if the time is not specified for a particular date, all jobs for that date are qualified.

The *time* parameter is one of the following:

month day year [, *hour : minute*]
day month year [, *hour : minute*]
hour : minute

in the format:

MMM dd yyyy [, *hh : mm*]
dd MMM yyyy [, *hh : mm*]
hh : mm

The DATE FROM *time* THRU *time* form can be used to bracket a range, such as in:

DATE FROM DEC 16 1988, 0:0 THRU 29 DEC 1988

FORM *form-name*

selects all jobs that have this form name specified.

LOC #*group*[.*destination*]

selects all jobs that have this location.

OWNER [*group-name.user-name* | *group-number,user-number*]

specifies all jobs with the specified owner. This value defaults to the creator accessor ID of the user. OWNER can be either group and user name or group and user number.

PAGES { > | < } *pages*

selects all jobs whose length in number of pages falls within the range specified. The PAGES subcommand allows two special cases. If PAGES > 0 is specified, all jobs are qualified. If PAGES < 0 is specified, jobs with zero pages are qualified. This makes PAGES < 0 equivalent to PAGES < 1.

REPORT *name*

selects all jobs with a matching report name. The report name is the name assigned to a job in the spooler, usually the group and user name of the job owner.

STATE *job-state*

selects all jobs in one of the following states: open, ready, hold, or print.

STATUS [/ OUT *filename* /] [DETAIL]

displays the status of all fonts that are defined in the spooler.

OUT *filename*

indicates where Spoolcom is to write the job status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

DETAIL

requests a complete list of font attributes. This parameter displays the attributes of fonts in the spooler subsystem for the specified *font-name* or for all fonts in the spooler.

Considerations for All Users

- All users can display the status of a job that is associated with a specific font name.
- If *font-name* is not specified, Spoolcom displays the status of all font jobs in the spooler.
- If *font-name* is specified, Spoolcom displays the status of the jobs associated with *font-name*.

- The FONT STATUS command without the DETAIL option lists all font jobs under the following headings:

```
FONT          JOB
```

- The FONT STATUS command with the DETAIL option lists each font job in the following format:

```
FONT: font-name
```

```
JOB: job-number
```

- Multiple jobs sent for printing to a location configured with a font job have the font printed before each print job. For example, assume a printer configured as follows:

```
Printer          Location
DEV $P1, EXCLUSIVE OFF LOC #A.NOFONT, DEV $P1
                  LOC #B.FONT, FONT ABC, DEV $P1
```

Assume you send print job 1 through #A.NOFONT and print jobs 2 and 3 through #B.FONT. The data sent to the printer is:

```
Job 1 data
Font ABC data
Job 2 data
Font ABC data
Job 3 data
```

- Multiple copies specified for a print job sent to a location configured with a font job have the font job printed only before the first copy. For example, assume the following printer configuration:

```
Printer          Location
DEV $P2, EXCLUSIVE OFF LOC #C.FONT, FONT ABC, DEV $P2
```

Assume you send two copies of print job 4 and one copy of print job 5 through #C.FONT. The data sent to the printer is:

```
Font ABC data
Job 4 data
Job 4 data
Font ABC data
Job 5 data
```

The font job data is printed only once before the first copy of print job 4. A font job is downloaded to a printer only when the device is opened by a location associated with a font.

Considerations for Authorized Users

- Only authorized users can define a special control job (*font-name*) in the spooler. The special control job might consist of downloadable VFC commands or possibly a special header page.
- Only authorized users can delete a special control job (*font-name*) with the FONT DELETE command.
- A font can be deleted even if it is associated with a location. When this occurs, the location ignores the font.
- Do not configure a device with DEV EXCLUSIVE OFF! if it is associated with locations that are configured with font jobs to affect the printing of a subsequent print job. The EXCLUSIVE OFF! attribute causes the device to be closed and opened between printing the font job and printing the subsequent print job, thereby terminating the effects of the font job. Configure a device in this situation with either EXCLUSIVE ON or EXCLUSIVE OFF.
- The maximum number of font jobs allowed for a spooler is 255. The FASTP print process, however, provides for nearly an unlimited number by allowing up to 4095 sections in a font job, where each section can apply to different locations. Sectioning is achieved by using a CONTROLBUF directive as described in Section 5, FONT Utility.

HELP Command

The HELP command displays the syntax of the Spoolcom commands.

```
HELP [ / OUT filename / ] [ command | ALL ]
```

OUT filename

specifies the file to which the syntax is written. If omitted, the syntax description is written to the home terminal.

If *OUT filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

command

is any Spoolcom command name. Spoolcom responds by displaying the detailed syntax of that command.

Omitting both *command* and ALL causes the abbreviated syntax of all commands to be displayed.

ALL

displays the detailed syntax of all commands.

Examples

1. To display on your terminal a list of all Spoolcom commands, enter
)
HELP
2. To display on your terminal the syntax information about the Spoolcom COLLECT command, enter
)
HELP COLLECT
3. To send to printer \$LP1 detailed syntax information about the Spoolcom JOB command, enter
)
HELP /OUT \$\$.#LP1/ JOB
4. If you want to display syntax information about the DEV command, enter
)
HELP DEV

JOB Command

If you are an authorized user, you can use the JOB command to alter attributes and change the status of any job.

Refer to the *Guardian User's Guide* for a description of job states and default attributes.

All users can perform these operations on their own jobs.

```
JOB [ job-number | ( qualifiers ) ] [ , subcommand ] ...
```

job-number

is the job number of the job.

qualifiers

specify the conditions that a job must meet. *qualifiers* is any number or combination of the following qualifiers, separated by commas and all enclosed in parentheses. Omitted qualifiers are not considered in the selection of jobs. Only one occurrence of each qualifier is allowed. If neither *job-number* nor *qualifiers* is specified, the spooler displays the status of all jobs on the system.

COLLECT *\$process-name*

selects all jobs collected by this collector. See [Appendix H, Migration Considerations for Pre-D41 Spooler Versions](#) for an example.

DATAFILE *data-filename*

selects all jobs with this data filename. If the volume and subvolume filename specifications are omitted, the default volume and subvolume are used. The filename must be specified and has no default. See [Appendix H, Migration Considerations for Pre-D41 Spooler Versions](#) for examples.

DATE { FROM *time* [THRU *time*] | THRU *time* }

selects all jobs created at the specified date and time or within the specified range of dates and times. If the date is omitted, today's date is used; if the time is not specified for a particular date, all jobs for that date are qualified.

The *time* parameter is one of the following:

```
month day year [ , hour : minute ]
day month year [ , hour : minute ]
hour : minute
```

in the format:

```
MMM dd yyyy [ , hh : mm ]
dd MMM yyyy [ , hh : mm ]
hh : mm
```


The DATE FROM *time* THRU *time* form can be used to bracket a range:

```
DATE FROM 16 DEC 1990, 0:0 THRU 29 DEC 1990
```

FORM *form-name*

selects all jobs that have this form name specified.

LOC #*group*[.*destination*]

selects all jobs that have this location.

OWNER [*group-name.user-name* | *group-number,user-number*]

selects all jobs created by the user with the specified user ID. If OWNER is specified without a user ID, it defaults to the creator accessor ID of the user. OWNER can be either group and user name or group and user number.

PAGES { > | < } *pages*

selects all jobs whose length in number of pages falls within the range specified. The PAGES subcommand allows two special cases. If PAGES > 0 is specified, all jobs are qualified. If PAGES < 0 is specified, jobs with zero pages are qualified. This makes PAGES < 0 equivalent to PAGES < 1.

REPORT *name*

selects all jobs with a matching report name specified. The report name is the name assigned to a job in the spooler, usually the group and user name of the job owner. It is printed on the banner page of a job.

The * and ? symbols provide report name selection through pattern matching. Replace one or more characters in the name with a symbol. The * symbol matches 0 or more characters. The ? symbol matches only one character.

STATE *job-state*

selects all jobs in one of the following job states: open, ready, hold, or print.

subcommand

is one of the subcommands described below. If no *subcommand* is specified, then STATUS is assumed. Unauthorized users can use these subcommands to change the attributes and status of their own jobs only. See [Table 4-9](#) for a list of which subcommands are valid with which job states.

COLLECT *\$process-name*

specifies that the job should be transferred to the collector *\$process-name*. The job must be in the HOLD state. If the job has a current, active collector, it is deleted from the current collect after the transfer completes successfully. If the job does not belong to a collector that is active, the job is not deleted from

the current data file. If the transfer fails, the job remains on the current data file and/or collector.

COPIES *number-of-copies*

specifies the number of copies to be printed. A job must be in the hold state before its copies attribute can be changed. The range for *number-of-copies* is 1 through 32767. The default is 1.

DELETE [!]

deletes a job from the spooler.

If you enter a JOB DELETE command on a job that is currently printing, Spoolcom might indicate that the job has been deleted before it actually is. This is because the supervisor can delete the job only after the print process has stopped printing the job.

If a *qualifier* list is used with the DELETE subcommand, the user must verify each deletion. Enter Y or y to delete a job.

If DELETE ! is entered, the user is not given the opportunity to verify each deletion. When deleting by job number, the ! has no effect.

FORM [*form-name*]

specifies a form name for the job. A job must be in the hold state before its form name can be changed. A form name is composed of up to 16 letters, numbers, and blanks.

The default name is all blanks.

HOLD

places the job in the hold state. A job must be in the hold state before most attributes can be modified.

To remove a job from the hold state, use the START subcommand.

HOLDAFTER [OFF | ON]

sets the hold-after-printing flag on or off. HOLDAFTER or HOLDAFTER ON causes the job to be placed in the hold state after it finishes printing. HOLDAFTER OFF (the default specification) allows the job to be deleted after printing.

LOC [*#location*]

specifies a new location for the job. The job must be in the hold state before its location attribute can be changed. If the *#location* parameter is omitted, then #DEFAULT is the new location.

OWNER { *group-name.user-name* | *group-number,user-number* }

designates a new owner of a job. You can use the local form of either the group and user name or the group and user number. The range for *group-number* and *user-number* is 1 through 255.

REPORT [*report-name*]

changes the report name for the job. A job must be in the hold state before its report name can be changed. A *report-name* is composed of as many as 16 letters, numbers, and blanks, and it must begin with a letter. It prints on the header page, 8 characters to a line.

The default report name is the user ID, that is, *group-name.user-name*.

SELPRI *selection-priority*

specifies the selection priority of the job. A job must be in the hold state before its priority attribute can be changed. When a job is added to the spooler, it is given a default priority of 4. The priority can be a value in the range of 0 through 7. Jobs print in order of priority.

START

takes a job out of the hold state and places it in the ready state.

STATUS [/ OUT *filename* /] [DETAIL]

displays the status of selected jobs.

OUT *filename*

indicates where Spoolcom is to write the job status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

DETAIL

requests a complete list of job attributes. This parameter displays the attributes of jobs in the spooler subsystem.

If *job-number* is not specified, then the status of all jobs in the spooler subsystem is returned.

[Table 4-9](#) shows the job states for which each JOB subcommand is valid. Job states are further described in the *Guardian User's Guide*.

Table 4-9. Job States During Which JOB Subcommands Can Be Used

JOB Subcommand	Job State			
	HOLD (H)	PRINT (P)	READY (R)	OPEN (O)
COPIES	H			
DELETE	H	P	R	
FORM	H			
HOLD	H	P	R	
HOLDAFTER	H	P	R	O
LOC	H			
OWNER	H			
REPORT	H			
SELPRI	H			
START	H			
STATUS	H	P	R	O

Considerations

- If *job-number*, *qualifiers*, or *subcommand* is not specified, then Spoolcom displays the status of all jobs.

JOB STATUS Display

The STATUS subcommand produces a display with the following headings:

```
JOB BATCH STA FLAGS OWNER TIME COPY PAGE REPORT LOCATION
```

[Table 4-10](#) describes the JOB STATUS display entries.

Table 4-10. JOB STATUS Display Entries (page 1 of 2)

Entry	Description
JOB	Is the job number.
BATCH	Is the batch number. If the job is not part of a batch job, this field is blank.
STA	Is the state of the job:
RDY	Spooler has finished collecting; job is queued and waiting to print.
HLD	The hold-before-printing flag is on or the hold-after-printing flag is on and the job has printed.
OPN	Job is still being collected by the spooler.
PRT	Job is currently printing.

Table 4-10. JOB STATUS Display Entries (page 2 of 2)

Entry	Description
FLAGS	is any or all of the following: 0-7 Job selection priority. A Hold after on. B Hold on (but job is still open). X Job is bad (examples: TGAL abended while spooling the job, or the job was spooled at level 3, but the collector process was closed before calling SPOOLEND).
OWNER	Is the job owner in the form <i>group-number,user-number</i> .
TIME	Is the month and day of job creation, that is, when the job was sent to the spooler. If it was today, the time is given in <i>hour:minute</i> form instead.
COPY	Is the number of copies to be printed.
PAGE	Is the number of pages in the job.
REPORT	Is the job report name.
LOCATION	Is the location assigned to the job.

Status of a Single Job

If the status of a single job is requested and the job is in the ready or print state, the following display is also shown:

```
JOB  LOCATION          DEVICE                SEQ  COPY  PAGE
```

[Table 4-11](#) describes the JOB STATUS display entries for a single job.

Table 4-11. JOB STATUS Display Entries—Non-Detailed Single Job Status

Entry	Description
JOB	Is the job number.
LOCATION	Is the complete location (#group and destination) for the job.
DEVICE	Is the name of the device that will print or is printing the job.
SEQ	Is the job sequence number in the device queue. PRINT is displayed if the job is printing.
COPY	Is the number of copies to be printed.
PAGE	Is the number of pages in the job.

Detailed Status of a Single Job

[Table 4-12](#) describes the JOB STATUS display entries when the detailed status for a single job is requested.

Table 4-12. JOB STATUS Display Entries—Detailed Single Job Status

Entry	Description
JOB	Is the job number.
STATE	Is the state of the job.
LOCATION	Is the location assigned to the job.
FORM	Is the form name, if any.
REPORT	Is the report name, if any.
HOLD BEFORE PRINT	Is yes or no.
HOLD AFTER PRINT	Is yes or no.
ABNORMAL	Is yes or no (see FLAGS X in Table 4-9).
SELECTION PRIORITY	Is the priority of the job (the same as the FLAGS number shown above).
PAGE SIZE	Is the number of print lines for each physical page.
CREATOR ACCESS ID	Is the job owner in the form <i>group-number,user-number</i> .
COPIES	Is the number of copies to be printed.
PAGES	Is the estimated number of pages in the report.
TOTAL LINES	Is the estimated total number of effective lines in the report.
OPEN TIME	Is the time when the spooler began spooling this file.
CLOSE TIME	Is the time when the spooler finished spooling this file.
DATA FILE	Is the name of the collector data file.
COLLECTED BY	Is the name of the collector.
UNITS ALLOCATED	Is the amount of disk memory used.
BATCH ID	Is the number of the spooler batch job.
BATCH NAME	Is the name for the batch job report.
MAXIMUM LINES	Is the maximum number of lines in effect for the report. The default value is no maximum.
MAXIMUM PAGES	Is the maximum number of pages in effect for the report. The default value is no maximum.

Examples for All Users

1. You can enter Peruse to find the job number of your spooled job, or you can use this Spoolcom equivalent:

```
)JOB (DATE FROM 10:00 THRU 11:00, OWNER)
```

The following information is displayed:

```
JOB BATCH STA FLAGS OWNER TIME COPY PAGE REPORT
LOCATION
3000 RDY 4 4,19 10:18 1 3 COMMSW RJ #LOC1
```

2. Then you can use the JOB command with the job number to get job status information:

```
)JOB 3000
```

The following information is displayed:

```
JOB BATCH STA FLAGS OWNER TIME COPY PAGE REPORT
LOCATION
3000 RDY 4 4,19 10:18 1 3 COMMSW RJ #LOC1

JOB LOCATION DEVICE SEQ COPY PAGE
3000 #LOC1.DEFAULT $DEV1 1 1 3
```

3. The DETAIL parameter gives even more job information:

```
)JOB 3000, STATUS DETAIL
```

The following information is displayed:

```
JOB: 3000
STATE: READY
LOCATION: #LOC1
FORM:
REPORT: COMMSW   RJ
HOLD BEFORE PRINT: NO
HOLD AFTER PRINT: NO
ABNORMAL: NO
SELECTION PRIORITY: 4
PAGE SIZE: 60
CREATOR ACCESS ID: 4, 19
COPIES: 1
PAGES: 3
TOTAL LINES: 169
OPEN TIME: 28 MAY 96, 10:23:00
CLOSE TIME: 28 MAY 96, 10:23:18
DATA FILE: $KIRK.TESTLIM.CD1
COLLECTED BY: $COL1
UNITS ALLOCATED: 2
BATCH ID:
BATCH NAME:
MAXIMUM LINES:
MAXIMUM PAGES:
```

4. In the following example, the JOB command is used to alter job attributes. The job is placed in the hold state to make the changes, the hold-after-printing flag (A) is set on, the report name for the header page is altered, and a status display is again requested, followed by the START subcommand to remove the job from the hold state:

```
)JOB 36, HOLD, HOLDAFTER, REPORT PB REPT, STATUS, START
```

The following information is displayed:

```
JOB BATCH STA  FLAGS OWNER TIME  COPY PAGE REPORT  LOCATION
36          HLD  4 A   4,19 10:18 1    5   PB REPT #UNTIL
```

5. Placing a job in the hold state truncates the display, as shown above. You must specify the STATUS subcommand after removing the hold in order to see the two-line display:

```
JOB BATCH STA  FLAGS OWNER TIME  COPY PAGE REPORT  LOCATION
36          RDY  4 A   4,19 10:18 1    1   PB REPT #UNTIL
```

```
JOB  LOCATION                DEVICE      SEQ  COPY  PAGE
36  #UNTIL.DEFAULT           1          1    1    1
```


6. The * pattern-matching symbol used in the following REPORT subcommand selects these jobs.

```
)JOB (REPORT su*)
JOB  BATCH STA  FLAGS OWNER    TIME    COPY PAGE REPORT
1803          RDY 4      255,255 10:49 1     1   SUPER  SUPER
2026          RDY 4      255,255 21:44 1     3   SUPER  SUPER
2027          RDY 4      255,220 21:47 1     3   SUPER  SPOOLER
2028          RDY 4      255,255 21:50 1     3   SUPER  SUPER
2029          RDY 4      135,120 21:51 1     4   SUPPORT GUEST
2030          RDY 4      255,220 21:52 1     4   SUPER  SPOOLER
2038          OPN 4      255,255 23:30 1     0   SUPER  SUPER
```

Examples for Authorized Users

Authorized users can perform any of the functions described above on any spooled jobs, not just their own.

LOC Command

The LOC command defines and modifies the spooler routing structure. Refer to Section 2 for a description of default location attributes and a description of the spooler routing structure.

All users can display the status of the spooler routing structure (with the STATUS subcommand) or produce a cross-reference list of locations, devices, and print processes (with the XREF subcommand).

In addition, authorized users can specify broadcast modes (BROADCAST), delete a location (DELETE), and connect or disconnect a device from a location (DEV).

```
LOC [ location ] , [ subcommand ]
```

location

is the logical destination of a job. If a print device is associated with a specific *location*, that print device becomes the physical destination of the job. The *location* parameter is a two-part name: *#group.dest*, as described under [Routing Structure](#) on page 2-16. If either name part is omitted, it takes its default value.

subcommand

is one of the subcommands described below.

```
BROADCAST [ OFF | ON ]
```

specifies the broadcast mode of the group. Only authorized users can use the BROADCAST subcommand.

BROADCAST or BROADCAST ON causes a job routed to *#group* to be printed on all devices connected to *#group*.

BROADCAST OFF (the default value) causes jobs routed to the group to be printed on the device (connected to the group) that can print the job the fastest.

```
DELETE
```

deletes from the spooler the entire group or the particular destination within the group. This works only if there are no jobs currently in the location being deleted. Only authorized users can use the DELETE subcommand.

```
DEV [ device-name ]
```

connects a location to or disconnects a location from a device.

If *#group.dest* is specified, the command refers to that particular group and destination.

If only *dest* is present, then the command refers to all existing locations with a group that has that destination in it.

If *device-name* is present, the command establishes a connection; if *device-name* is absent, any existing connection with a device is broken.

Only authorized users can use the DEV subcommand.

FONT [*font-name*]

connects a location to or disconnects a location from a font. Only authorized users can use the LOC FONT subcommand.

If *#group.dest* is specified, the command refers to that particular group and destination.

If only *dest* is present, the command refers to every group that has that destination in it.

If *font-name* is present, this font will be downloaded before the execution of any jobs that are sent to this location. If *font-name* is absent, any font associated with this location will be removed.

For considerations about when font data is printed, see the [FONT Command](#) on page 4-42.

STATUS [/ OUT *filename* /] [DETAIL]]

displays the attributes of locations in the spooler subsystem. If the XREF, BROADCAST, DELETE, DEV, and FONT subcommands are not specified, the default subcommand is STATUS.

OUT *file-name* indicates where Spoolcom is to write the location status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

DETAIL requests a complete list of all location attributes. You must specify the keyword STATUS if you use the DETAIL parameter.

If both *#group* and *dest* are specified, the status of only that particular location is returned. If *dest* is not specified, the status of all destinations associated with the group is returned. If *#group* is not specified, the status of all locations in the routing structure is returned.

See [Considerations](#) on page 4-60 for more information about the STATUS subcommand.

XREF [/ OUT *file-name* /]

produces a cross-reference list of locations, devices, and print processes ordered by location. If a location is not specified, a complete cross-reference of all locations is produced. If a location is specified, it can be entered as:

#group | *#group.dest*

If *#group* is specified, all locations within that group are listed. If *#group.dest* is specified, a cross-reference for that location is produced.

If OUT *filename* is not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

Considerations

- Do not use the BROADCAST subcommand until you have defined at least one of the broadcast destinations. If a LOC *location*, BROADCAST command is issued before a destination in the group has been defined, the spooler will automatically define the *#group.DEFAULT* location. If the DEFAULT destination is really part of the broadcast group (it does not have a device defined), jobs sent to the group will remain in the READY state after they have been printed on the defined destinations.
- The following points apply to the STATUS subcommand:
 - If neither *location* nor a subcommand is specified, then Spoolcom displays the status of all locations.
 - An user can obtain the status of any location in the spooler subsystem.
 - Using the STATUS subcommand without specifying *#group* or *dest* produces a display with the following headings for all spooler locations:

LOCATION	FLAGS	DEVICE	FONT
----------	-------	--------	------

Using the STATUS subcommand with *#group* or *dest* produces a two-line display with the following headings for each job at that location:

LOCATION	FLAGS	DEVICE	FONT
JOB LOCATION		DEVICE	SEQ COPY PAGE

Specifying LOC DETAIL or LOC STATUS DETAIL produces another display:

```
LOCATION:
  BROADCAST:
  DEVICE
  FONT NAME:
```

[Table 4-13](#) describes the LOC STATUS display entries.

Table 4-13. LOC STATUS Display Entries

Entry	Description
LOCATION	Is the # <i>group.dest</i> whose status is being displayed.
FLAGS	Displays a B if broadcast is on.
DEVICE	Is the device associated with the location, if any.
FONT	Defines a special control job in the spooler. The special control job might consist of downloadable vertical forms control (VFC) command or possibly a special header page.
JOB	Is the job numbers of all jobs at that destination.
SEQ	Is the job position in that destination queue.
COPY	Is the number of copies to be printed.
PAGE	Is the number of pages in that job.
BROADCAST	Is ON or OFF.
FONT NAME	Identifies the particular font associated with this location.

Examples for All Users

1. If you send your job to a real location, as in this example, the XREF option reveals the device name and print process name associated with it:

```
)LOC #LPR1, XREF
```

The following information is displayed:

LOCATION	DEVICE	PRINT PROCESS
#LPR1.DEFAULT	\$LP1	\$SPLA

2. If you send your job to a location that does not exist, there is no device or print process associated with that fictitious location:

```
)LOC #LOOK, XREF
```

Examples for Authorized Users

1. The following command connects device \$LP to the location #PRIN.DEFAULT:

```
)LOC #PRIN.DEFAULT, DEV $LP
```

2. The following command deletes location #LP8 from the spooler:

```
)LOC #LP8, DELETE
```

OPEN Command

The OPEN command specifies the spooler supervisor with which Spoolcom communicates.

```
OPEN [ \system.$supervisor ]
      [ $supervisor           ]
      [ \system                ]
```

\system

is the name of a system (node) where the spooler supervisor process resides. The default *\system* is the current system.

\$supervisor

is the process name of a supervisor in remote or local form. The default process name for *\$supervisor* is *\$SPLS*. When opening a remote supervisor, the system name can be specified without the process name. (For example, if you specify *\FAR* as the system name, Spoolcom opens the process *\FAR.\$SPLS*.)

Consideration

- When starting Spoolcom, you can specify the name of the supervisor to open. If no supervisor name is specified, then Spoolcom opens a process named *\$SPLS* (the default name). After Spoolcom is running, you can use the OPEN subcommand to open a different supervisor.

Example

You can use this command to communicate with a job you have spooled to a remote location, for example, supervisor *\$SPFAR* on network *\FARSYS*:

```
) OPEN \FARSYS.$SPFAR
```

PRINT Command

If you are an authorized user, you can use the PRINT command to specify attributes and obtain the status, and change the status of the spooler print processes. See [PRINT STATUS Display](#) on page 4-67 for a description of the print-process states.

All users can obtain the status of any spooler print processes.

```
PRINT [ $process-name ] [ , subcommand ] ...
```

\$process-name

is the name of the print process to which the subcommands refer.

All subcommands except STATUS require that this parameter be included.

subcommand

is one of the subcommands described below. If *subcommand* is not specified, then STATUS is assumed.

BACKUP *backup-cpu*

is the number of the processor (-1 through 15) that is to run the backup print process. The default number is -1, which specifies that the print process has no backup.

Tandem print processes do not run as process pairs, so this subcommand enables the spooler supervisor to restart a print process. If *backup-cpu* is specified and a print process primary processor goes down, the spooler supervisor restarts the print process and any of its devices for which DEV RESTART is set ON. A stopped process is not restarted. When the print process is restarted, it is restarted in the backup processor, and the values for CPU and BACKUP are switched. The operation of user-written print process pairs is not affected by this switch.

CPU *cpu*

is the number of the processor (0 through 15) that runs the print process. The default processor is the same processor as the supervisor.

DEBUG [OFF | ON]

sets the debug mode of the print process.

DEBUG or DEBUG ON specifies that this print process runs in debug mode; DEBUG OFF (the default) specifies that the print process does not run in debug mode. See the information on debugging print processes in the *Spooler Programmer's Guide*, and see [Considerations](#) on page 4-65

DELETE

removes the print process from the spooler subsystem. The print process must be in the dormant state before you can delete it. See [Considerations](#) on page 4-65 for more information on putting a print process in a state so that you can delete it. Also, all devices associated with this process must be disconnected from this print process before you can delete it.

FILE *program-filename*

specifies the program file for this print process. The print process provided by Tandem is `$SYSTEM.SYSTEM.FASTP`.

If omitted, the supervisor assumes that the process is an independent print process (a print process that is already running when the supervisor is started). Refer to the *Spooler Programmer's Guide* for more information on independent print processes.

PARM *parameter*

is a print-process parameter that is passed by the supervisor to the print process in the Startup message. Its meaning is defined by a user-written print process. The Tandem print process, FASTP, does not use this parameter. The range of values is -32768 through +32767. The default is 0.

PRI *execution-priority*

specifies the execution priority of the print process. The range for this parameter is 1 through 199. The default is 145.

START

takes a print process out of the procerror state after the cause of the failure has been identified and fixed.

STATUS [/ OUT *filename* /] [DETAIL]

displays the attributes of print processes in the spooler subsystem. If a print process is not specified, then the status of all print processes in the spooler subsystem is returned. If a print process is specified, then the status of only that print process is returned.

OUT *filename*

indicates where Spoolcom is to write the print-process status. If not specified, the status is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file

(file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

DETAIL

requests a complete list of all the print-process attributes.

XREF [/ OUT *filename* /]

produces a cross-reference listing of print processes, devices, and locations, ordered by print process. If a print process is not specified, a complete cross-reference is produced. (This more time than any other XREF command.)

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

Considerations

- If neither $\$process-name$ nor *subcommand* is specified, then Spoolcom displays the status of all print processes.
- Any user can use the STATUS subcommand to obtain the status of print processes in the spooler subsystem.
- Only members of the authorized group can initialize and set the attributes of a print process.
- Attributes of a print process can be specified only when the print process is in the dormant or procerror state.
- The supervisor runs print processes only as needed. A print process controlling an exclusive device runs all the time, while a print process controlling only shared devices runs when the print process is actually printing a job on one of its devices.
- Before you can delete a print process, you might need to perform the following operations to put the process in the dormant state:
 - For the devices that use the print process, set DEV $\$device$ EXCLUSIVE OFF so the print process will close the device when no more jobs are waiting to be printed.
 - Drain the print process by draining all the devices that use the print process. The DEV $\$device$ DRAIN command drains a device. The print process should become dormant approximately two minutes after the process prints its last job. If the print process does not become dormant at this time, issue a STOP command from a TACL prompt to stop the process.

- When a print process is in debug mode, it is not timed out by the spooler. This means that the spooler waits indefinitely for a response. For this reason, print processes should never be debugged on a production spooler.

PRINT STATUS Display

The STATUS subcommand produces a display with the following headings:

```
PRINT      STATE          FLAGS  CPU      PRI
```

[Table 4-14](#) describes the PRINT STATUS display entries.

Table 4-14. PRINT STATUS Display Entries (Without DETAIL Option)

Entry	Description						
PRINT	Is the name of the print process whose status is being given.						
STATE	Is the state of the print process. STATE can be one the following: <table border="0" style="margin-left: 20px;"> <tr> <td>ACTIVE</td> <td>The print process is running. This means that it is printing a job, it is controlling a device exclusively, or it is an independent print process. Some print processes remain active for a short time after printing the last job in the queue before becoming dormant. This delay avoids additional startup time if a job enters the print queue during this period.</td> </tr> <tr> <td>DORMANT</td> <td>The print process is not running. This means it has no job to print, controls only shared devices, and is not an independent print process.</td> </tr> <tr> <td>ERROR %<i>nnn</i></td> <td>The print process has failed. It must be removed from this state before it can become active again. The octal number that follows ERROR indicates the error condition. It can be either %1000 plus a file-system error number or %100000 plus a NEWPROCESS error number.</td> </tr> </table>	ACTIVE	The print process is running. This means that it is printing a job, it is controlling a device exclusively, or it is an independent print process. Some print processes remain active for a short time after printing the last job in the queue before becoming dormant. This delay avoids additional startup time if a job enters the print queue during this period.	DORMANT	The print process is not running. This means it has no job to print, controls only shared devices, and is not an independent print process.	ERROR % <i>nnn</i>	The print process has failed. It must be removed from this state before it can become active again. The octal number that follows ERROR indicates the error condition. It can be either %1000 plus a file-system error number or %100000 plus a NEWPROCESS error number.
ACTIVE	The print process is running. This means that it is printing a job, it is controlling a device exclusively, or it is an independent print process. Some print processes remain active for a short time after printing the last job in the queue before becoming dormant. This delay avoids additional startup time if a job enters the print queue during this period.						
DORMANT	The print process is not running. This means it has no job to print, controls only shared devices, and is not an independent print process.						
ERROR % <i>nnn</i>	The print process has failed. It must be removed from this state before it can become active again. The octal number that follows ERROR indicates the error condition. It can be either %1000 plus a file-system error number or %100000 plus a NEWPROCESS error number.						
FLAGS	Can be either or both of the following entries: <table border="0" style="margin-left: 20px;"> <tr> <td>I</td> <td>The process is an independent process.</td> </tr> <tr> <td>D</td> <td>The process is in debug mode.</td> </tr> </table>	I	The process is an independent process.	D	The process is in debug mode.		
I	The process is an independent process.						
D	The process is in debug mode.						
CPU	Gives the numbers of the processors that will run the primary and backup processes.						
PRI	Is the execution priority at which the print process runs.						

When DETAIL is specified, the information in [Table 4-15](#) is shown:

Table 4-15. PRINT STATUS Display Entries (With DETAIL Option)

Entry	Description
PRINT PROCESS	Is the name of the print process whose status is being given.
STATE	Is the same as that shown in Table 4-14 except when an error has occurred. Then only the word ERROR is displayed and the error number is displayed in LAST ERROR.
LAST ERROR	Is the octal number of the last error printed on the error log file. It can be %1000, %2000, %3000, %4000, or %5000 plus a file-system error number, or %100000 plus a NEWPROCESS error number. The octal error number is a 16-bit value where bits 8 through 15 contain the file-system error number.
DEBUG	Indicates that the process is in debug mode.
INDEPENDENT	Indicates that the process is an independent process.
PROGRAM FILE	Is the name of the print process program file.
CPU BACKUP	Give the numbers of the processors that will run the primary and backup processes.
PRIORITY	Is the execution priority at which the print process runs.
PARM	Is the print-process parameter.

Examples for All Users

1. To display the status of a print process, enter

```
) PRINT $SPLA
```

The following information is displayed:

```
PRINT    STATE          FLAGS  CPU    PRI
$SPLA    ACTIVE          5, 9  145
```

2. To display the status of that same print process in greater detail, enter

```
) PRINT $SPLA , STATUS DETAIL
```

The following information is displayed:

```
PRINT PROCESS: $SPLA
STATE: ACTIVE
LAST ERROR: NONE
DEBUG: OFF
INDEPENDENT: NO
PROGRAM FILE: $SYSTEM.SYSTEM.FASTP
CPU: 5
BACKUP: 9
PRIORITY: 145
PARAM: 0
```

Examples for Authorized Users

You can use the PRINT command to define the print process \$XP to be a copy of the program file \$SYSTEM.SYSTEM.FASTP and to run on CPU 3:

```
)PRINT $XP, FILE FASTP, CPU 3
```

Then you can start the print process \$XP, moving it out of the procerror state, so it is ready to print jobs:.

```
)PRINT $XP, START
```

SPOOLER Command

If you are an authorized user, you can use the SPOOLER command to start, stop, and obtain the status of the spooler subsystem.

Refer to the *Guardian System Operations Guide* for a description of spooler states.

All users can obtain the status of the spooler subsystem with this command.

```
SPOOLER [ , subcommand ] ...
```

subcommand

is one of the subcommands described below. If *subcommand* is not specified, then STATUS is assumed.

DRAIN

brings the spooler subsystem to an orderly halt after all jobs that are currently printing or spooling have finished. This subcommand moves the spooler from the active to the dormant state.

DUMP *filename*

specifies that a snapshot of the spooler supervisor process be written to the designated file while the spooler is running. The *filename* parameter specifies the name of the file where the dump is to be written. The dump file cannot exist; the supervisor process creates a new file with file code 130, which is an Inspect save file that you can subsequently read by using the Inspect debugger.

The information dumped includes the user data space (including extended data segments); the values of the P, E, L, and S registers; the values of the register stack; and the names and status of any files opened by the supervisor process.

MGRACCESS [ON | OFF]

specifies whether group managers (user ID *n*, 255) can access jobs belonging to their group members.

MGRACCESS or MGRACCESS ON specifies that group managers can access those jobs. This capability applies to all group managers using the spooler.

MGRACCESS OFF (the default value) specifies that group managers cannot access those jobs.

START

starts the spooler (collectors and print processes). You can start the spooler when it is in the warm or cold state.

STATISTICS

displays additional statistics maintained with the spooler for the purpose of better understanding how the spooler is currently being used. See [SPOOLER STATISTICS Display](#) on page 4-72.

STATUS [/ OUT *filename* /] [DETAIL]

displays the status of the spooler subsystem. See [SPOOLER STATUS Display](#) on page 4-74.

OUT *filename*

indicates where Spoolcom is to write the status information. If not specified, the information is written to the Spoolcom OUT file (usually the home terminal).

If OUT *filename* is specified, Spoolcom writes the data to it in a manner suitable for the file type, process, or device. The *filename* parameter must be specified; if the file does not exist, Spoolcom creates an EDIT file (file type 101). The file can be an EDIT file, unstructured file, relative file, entry-sequenced file, spooler job file, process, or device.

DETAIL

requests a complete list of all status information.

Considerations

- Any user can obtain the status of the spooler; however, only members of the authorized group can modify the attributes of the spooler.
- The MGRACCESS subcommand setting does not persist through a warm start of the spooler. The default setting is MGRACCESS OFF; if you want access on, you must reenter MGRACCESS ON after a warm start.
- The DRAIN subcommand stops the spooler in an orderly manner. It is the only recommended way to stop the spooler. Following the SPOOLER DRAIN subcommand:
 1. The collectors allow current jobs to finish but reject new opens with a file-system error 66 (device downed). Each collector stops when it has no more open jobs.
 2. Each print process finishes printing any active jobs and then stops.
 3. After all collectors and print processes have stopped, the supervisor stops.
 4. The spooler enters the dormant state, ready to be warmstarted.

SPOOLER STATISTICS Display

You can use the STATISTICS subcommand to access additional spooler statistics. Much of the data is assembled after the report is requested. Because preparation of the report takes precedence over other spooler requests, using it often can affect spooler performance.

The STATISTICS subcommand produces a display similar to the following example:

```
)spooler,statistics

SPOOL SUPERVISOR:  $SPLS
JOBS                :104
OPEN JOBS           :1
SUPERVISOR OPENERS  :50
DEVICE QUEUE NODES :103
RECEIVE QUEUE ENTRIES :55
JOBS CURRENTLY PRINTING :54
```

[Table 4-16](#) describes the SPOOLER STATISTICS display entries.

Table 4-16. SPOOLER STATISTICS Display Entries (page 1 of 2)

Entry	Description
JOBS	Is the number of print jobs known to the spooler supervisor regardless of their state. This count can be as large as the maximum number of jobs allowed in the spooler (4095). Jobs early in this collection process, which are known only to the collector, are not included in the count. This entry is also reported with Event Message Service (EMS) threshold messages.
OPEN JOBS	Is the number of jobs known to the spooler supervisor that have not yet been fully collected. This count can be as large as the maximum number of jobs allowed in the spooler (4095). Jobs known only to the collector are not included in the count.
SUPERVISOR OPENERS	Is the number of opens to the supervisor. It is the sum of the number of opens from collector and print processes plus those from Spoolcom, Peruse, and any user applications that open the supervisor. This count can be as large as the maximum number of openers allowed by the supervisor (70). This entry is also reported with EMS threshold messages.

Table 4-16. SPOOLER STATISTICS Display Entries (page 2 of 2)

Entry	Description
DEVICE QUEUE NODES	<p>Represents the number of jobs queued for all devices. All jobs in the spooler take one or more device queue nodes, except jobs that are in the OPEN and HOLD states. If a group location or broadcast location is the destination of a job, the number of device queue nodes used by the job is the number of devices configured for the location. Jobs that are directed to <i>group.destination</i> or a location that is not assigned to a device use only one device queue node.</p> <p>In general, the number of queue nodes should not be a limit in the use of the spooler. However, if additional queue nodes cannot be obtained for a job and no other limits have been exceeded, the job is placed on hold.</p>
RECEIVE QUEUE ENTRIES	<p>Is the number of requests received by the supervisor; they are often queued in a receive queue to permit selective processing. This count is the current number of requests queued.</p> <p>The receive queue size is equal to the number of openers allowed in the supervisor, plus the maximum number of collectors, plus the maximum number of print processes, plus the maximum number of devices allowed: $70 + 15 + 45 + 255 = 385$.</p>
JOBS CURRENTLY PRINTING	<p>Is the number of jobs that have begun but have yet to finish printing. The maximum value for this count is the configured maximum number of devices.</p>

SPOOLER STATUS Display

The STATUS subcommand produces a display with the following headings:

```
SPOOLER          STATE          LOGGING FILE          LAST ERROR
```

[Table 4-17](#) describes the SPOOLER STATUS display entries.

Table 4-17. SPOOLER STATUS Display Entries

Entry	Description
SPOOLER	Is the process name of the spooler.
STATE	Is the state of the spooler (active, drain, warm, or cold). Spoolcom cannot communicate with a dormant spooler. Refer to the <i>Spooler Programmer's Guide</i> for a full description of the spooler states.
ACTIVE	The spooler is fully operational and ready to accept output from application processes.
DRAIN	The collector process has stopped accepting new jobs and will enter the dormant state after all open jobs have finished spooling. Once drained, the spooler enters the dormant state.
WARM	This state indicates the restarting of a spooler that was previously active.
COLD	This state indicates the starting of a new spooler.
LOGGING FILE	Is the spooler error log file.
LAST ERROR	Is the octal number of the last error sent to the error log file. Bits 8 through 15 contain the octal file-system error number. All of the file-system errors are described in the <i>Guardian Procedure Errors and Messages Manual</i> .

Example for All Users

- Using the SPOOLER command, you can obtain the status of your spooler:

```
) SPOOLER
```

The following information is displayed:

```
SPOOLER          STATE          LOGGING FILE          LAST
ERROR
$SPLS           ACTIVE          $0
```

2. Use the STATUS DETAIL command to determine whether manager access is available:

```
) SPOOLER, STATUS DETAIL
```

The following information is displayed:

```
SPOOL SUPERVISOR: $SPLS  
STATE: ACTIVE  
MGRACCESS: ON  
LOG FILE: $0  
LAST ERROR: NONE
```

Example for Authorized Users

You use the SPOOLER command to start the spooler, including any collectors and print processes associated with it, and to specify manager access if that feature is used at the site:

```
) SPOOLER, START, MGRACCESS ON
```


5 Font Utility

The Font utility program helps you create font jobs within the Tandem spooling system. A font job contains commands that can be sent to a printer to control such features as character sets, vertical forms-control specifications, or compressed printing. Font jobs, therefore, frequently contain both printable and nonprintable text, such as PCL escape sequences. With the Font utility, however, you enter all font descriptions in plain text; no nonprintable characters are used.

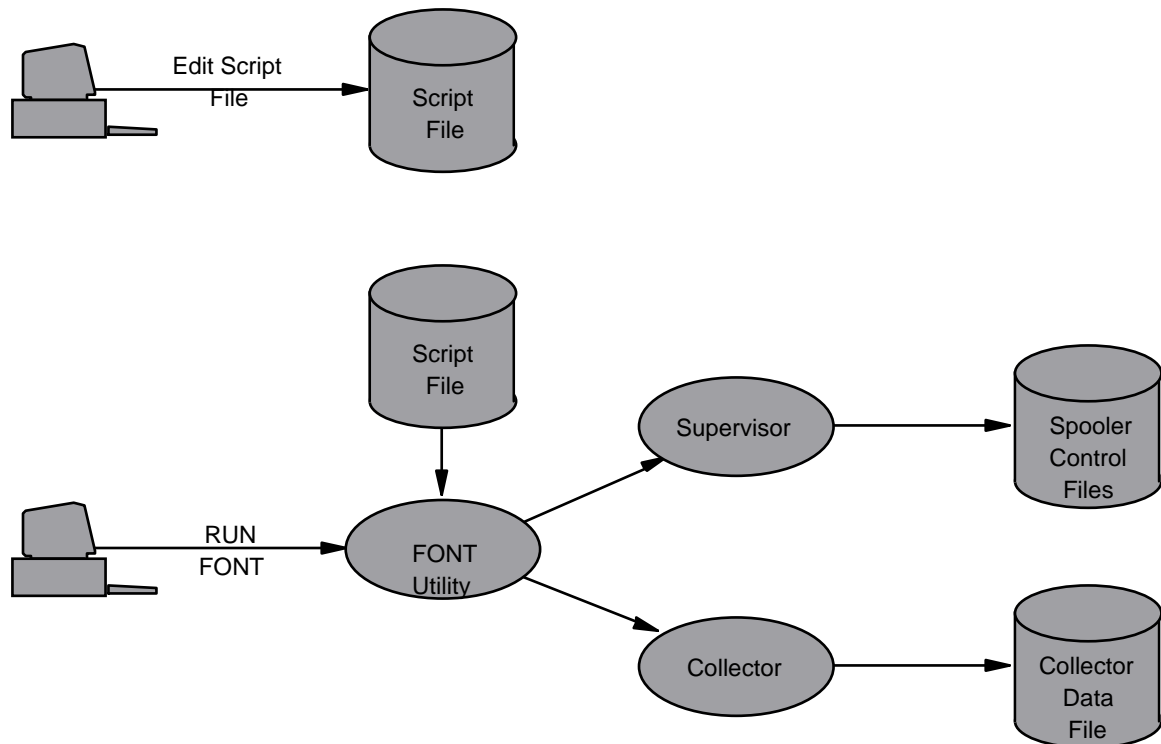
The following topics are described in this section:

- [How the Font Utility Works](#) on page 5-2
- [Running the Font Utility](#) on page 5-3
- [Script File Syntax](#) on page 5-5

How the Font Utility Works

To use the Font utility, you enter font job descriptions and locations in plain text in an EDIT file, called a script file, which the Font utility reads to create the font jobs and to associate them with spooler locations. Preparing the script file and running the Font utility is illustrated in [Figure 5-1](#).

Figure 5-1. Font Utility Operation



CDT 013CDD

The Font utility creates normal spooler jobs by sending data to a collector process. Then the Font utility makes these jobs into font jobs. As directed by script file text, the Font utility can create locations and link font jobs to locations by sending appropriate commands to the spooler supervisor.

Running the Font Utility

You can run the Font utility by entering the following command at the TACL prompt:

```
FONT / IN script-file [ , run-options ] ... / [ supervisor ]
```

IN script-file

designates an EDIT file containing details of the font jobs to add and locations to set up. See [Script File Syntax](#) on page 5-5 for the requirements of the file.

The *IN script-file* parameter is actually a run option. If the IN parameter is omitted, the Font utility runs interactively, but this is not recommended because the utility is not designed to accept data interactively.

run-options

are any run options for the command-interpreter RUN command, separated from each other by commas and set off in the command line by / characters.

The following run option might typically be used with the Font utility:

OUT list-file

indicates where the Font utility is to write status or error information. If not specified, the status is written to the OUT file in effect for the RUN command (usually the home terminal).

If *OUT list-file* is specified, the Font utility writes the data to it in a manner suitable for the file type, process, or device. The *list-file* parameter must be specified; it can be an unstructured file, entry-sequenced file, process, or device. Output to an EDIT file is not accepted.

For information on other run options, see the description of the RUN command in the *TACL Reference Manual*.

supervisor

is the name of the spooler supervisor with which the Font utility communicates. If *supervisor* is omitted, then the Font utility assumes that \$SPLS is the supervisor.

If you want to communicate with a remote supervisor, use the remote form of the supervisor process name (for example, \FAR.\$SPLS). Font and your terminal process should execute on the same node.

Considerations

- The collector process used by the Font utility is the first collector process in the active state that is configured in the spooler.
- Error messages are described in Appendix D.
- After loading a font job into the spooler, you can check its contents by using the `Peruse LIST` command. If you specify the octal and control modifiers, you can see the nonprintable characters and any embedded procedure calls. For example, the following command displays the entire contents of job 45:

```
55> PERUSE
  _JOB 45; LIST ALL O C
```

- Font jobs are not usually used to download fonts; rather, the font job is sent by the spooler to the device for every print job that actually prints.
- Font jobs typically control the following operations involved in printing:
 - Selecting paper from several input bins
 - Specifying page orientation, landscape or portrait
 - Specifying simplex or duplex printing
 - Selecting a different font (already installed in the printer)
 - Enabling transparent printing mode
- Because the Font utility modifies the spooler configuration, you must be the owner of the spooler or a system operator (user ID 255, *n*) to use this utility.

Example

The following example illustrates how to run the Font utility and shows output from the script file described in the [Script File Example](#) on page 5-17.

```
FONT /IN FSCRIPT / $SPLS
```

```
Welcome to the FONT loader
Spool supervisor = $SPLS
Collector = $S
FONT: PORTRAIT = JOB: 0369
FONT: TRANSP = JOB: 0370
Location: #LX.PORT Device: $LASER.#A Font: PORTRAIT
Location: #LX.GLASS Device: $LASER.#A Font: TRANSP
FONT loading is complete
```


Script File Syntax

A script file has two sections: the **font description** is a required section that describes font jobs; the **location** is an optional section that associates spooler locations with a particular font job.

The contents of the script file are first described by a skeleton, which shows required and optional directives. Following the skeleton are general rules for directives, directive descriptions, and examples.

Script File Skeleton

A skeleton of the script file syntax follows:

```
?START-FONT
{ ?FONT font-name
  [ font-description-entries ] ... } ...
?END-FONT
[ locations-section ]
font-description-entries are:
  normal-font-data
  ?CONTROL operation param
  ?CONTROLBUF operation data [ ; ]
  ?SETMODE function param1 param2
  ?SOURCE source-file [ beginning-count [ trailing-count ]
]
locations-section is:
  ?START-LOCATIONS
    { location-name [,] font-name [,] [ device-name ] }
  ...
  ?END-LOCATIONS
```

General Syntax Rules for Script File Directives

The following rules apply to directives in the script file:

- A directive is a line that has a question mark (?) as the first nonblank character.
- At least one space must separate directive parameters from the directive name and from other parameters.
- Directive names are not case-sensitive.
- A font job begins with the FONT directive and consists of the following elements in any quantity and in any order:

```
normal-font-data
?CONTROL
?CONTROLBUF
?SETMODE
```

START-FONT and END-FONT Directives

The START-FONT directive must be the first directive in the script file, because this directive starts the required font description section. The END-FONT directive terminates the font description section.

```
?START-FONT
.
.
.
?END-FONT
```

Considerations

- All text, including other directives, preceding the START-FONT directive in the file is treated as comments.
- There can be only one font description section in a script file. There can be descriptions of many fonts in that one section, however.
- Any text between an END-FONT directive and a START-LOCATIONS directive is ignored.

FONT Directive

The FONT directive designates a font name and begins the description of a particular font job. A font job contains commands that can be sent to a printer to control such features as character sets, vertical forms-control specifications, and compressed printing.

```
?FONT font-name
```

font-name

identifies the specific font name. The *font-name* parameter is 1 through 16 alphanumeric characters; the first character must be a letter. If the font already exists, the subsequent font job description entries modify the existing description. If the font does not exist in the spooler, a new font job is created.

Considerations

- The description of this font job continues until either the next FONT directive or the END-FONT directive occurs in the script file.
- If there are no font description entries between this FONT directive and either the next FONT or END-FONT directive, the font is considered empty. The Font utility does not create an empty font and does not replace an existing font with an empty font.
- If a font name associated with a job already exists in the spooler, the Font utility deletes the current font job and links the new font job number to this font name.

△ **Caution.** Make sure that the font name you specify is not an existing spooler font unless you intend to replace the existing font.

Normal Font Data

Normal font data includes commands that can be sent to a printer to control such features as character sets, vertical forms-control specifications, and compressed printing. Refer to your printer documentation for information about printer control commands. The format in which a command must appear in a script file is called a data element.

```
data-element [ , data-element ] ...
```

data-element

represents text for a particular instruction for a single font job and can include one or more lines of text. The four formats of *data-element* are

```
"actual-text"  
#byte-value
```

%byte-value
Hbyte-value

To enter actual text, delimit the text with quotation marks ("). To specify the value of a byte, precede *byte-value* with the symbol for the numeric base as follows:

Decimal
 % Octal
 H Hexadecimal

Considerations

- A single comma must separate data elements that appear on a single line.
- The Font utility inserts each line of data into the spooler font job with one input/output (I/O) operation.
- The Font utility does not add line feed (LF) or carriage return (CR) characters unless they are part of the data. Interpretation of this data by the spooler print process, input/output process (IOP), or printer depends on any SETMODE directives that precede the CONTROLBUF directive.
- To prevent each line being terminated with an LF CR by the print process, use a SETMODE directive (?SETMODE 6 0 1) to enable transparent operation.

Example

The following example is a font job that allows a Tandem 5537 or 5577 printer to print double-sided. The normal data appears between the two SETMODE directives.

```
?FONT LXPX2D           ! Font name
?SETMODE 6 0 1        ! Transparent
%33,"E"               ! Reset
%33,"(8U",%33,"(s0p16.67h7.0v0s0b8T" ! Rmn 8 Pres Elite
                                     ! 7 pt 16.67 cpi
%33,"&l00"             ! Portrait
%33,"&l26A"           ! A4 Paper
%33,"&k7.045H"         ! HMI = 7.045 120ths
%33,"&l3.9C"          ! VMI = 3.9 48ths
%33,"&l1S"            ! Duplex Long Edge Binding
%33,"&a1L"            ! Left Margin 1
?SETMODE 6 1 0        ! Normal
?CONTROLBUF 103      ! FASTP Top Of Form Substitute
%33,"&a67r0c1L";      ! Posn to Row 67 Col 0 Left
                                     ! Margin 1
?CONTROLBUF 103      ! FASTP Top Of Form Substitute
%33,"&a1L",%14,%15;   ! Left Margin 1, FF CR
!!
! Note that 5573 and 5577 printers require an optional font
! cartridge to supply the 7-point type font in this job.
```

CONTROL Directive

The CONTROL directive embeds a CONTROL procedure call in the font job. This directive controls operations on devices.

```
?CONTROL operation param
```

operation

is a number in the range 0 through 32767. This parameter corresponds directly to the *operation* parameter of the CONTROL procedure. See the CONTROL procedure in the *Guardian Procedure Calls Reference Manual* for *operation* values.

param

is a number in the range 0 through 32767. This parameter corresponds directly to the *param* parameter of the CONTROL procedure. See the CONTROL procedure in the *Guardian Procedure Calls Reference Manual* for *param* values. At least one space must separate *operation* from *param*.

Example

```
?CONTROL 1 0      ! Go to top of form (form feed)
```

CONTROLBUF Directive

The CONTROLBUF directive imbeds a CONTROLBUF procedure call in the font job.

```
?CONTROLBUF operation data [ ; ]
```

operation

is a value defined by the device, such as:

1 represents load DAVFU (printer subtype 4).

The value for *operation* must be a number in the range 0 through 32767. For information on how the FASTP print process uses this directive, see examples that follow.

data

is the data to be included in the CONTROLBUF procedure call. Data for a single font job can include one or more lines of text. The format for a line of data is

```
data-element [ , data-element ] ...
```

For the format for *data-element*, see [Normal Font Data](#), earlier in this section.

;

is a semicolon required to terminate CONTROLBUF data if any normal-text data follows. Otherwise, CONTROLBUF data can be terminated by a FONT, an END-FONT, or another font description directive.

Considerations

- CONTROLBUF parameter values are usually described in the documentation provided with your printer.
- CONTROLBUF data must conform to the same rules as normal font data.
- CONTROLBUF data is assembled in a buffer. The Font utility calculates the number of bytes supplied so that you do not need to specify buffer length in the Font utility script, as you would if you called the CONTROLBUF procedure directly.
- CONTROLBUF operation values have special meaning for FASTP, as follows:
 - 10 Loads a character translation table into FASTP; see [Example of User-Character Translation \(CONTROLBUF 101\)](#) on page 5-11.
 - 10 Loads substitute top-of-form sequences into FASTP; see [Example of Top-of-Form Substitution \(CONTROLBUF 103\)](#) on page 5-11.
 - 113 Allows you to create sections imbedded in a font job to be used by FASTP. These sections limit the scope of the font job data to one or more locations. The location names must be specified as the first data elements in the directive as follows:

```
CONTROLBUF 113 "location-name-1 "
              [ "location-name-n " ] ... ;
normal-font-data
```

Blank-fill location names to 16 characters in length. They are not case-sensitive. Terminate data (the specified location names) in a CONTROLBUF 113 directive with a semicolon (;).

The locations specified in one CONTROLBUF 113 directive apply either until the next CONTROLBUF 113 directive or until the END-FONT directive. See [Example of Virtual Font Jobs With FASTP \(CONTROLBUF 113\)](#), in the examples that follow.

Examples

The CONTROLBUF data follows the CONTROLBUF operation. The CONTROLBUF directives are terminated by semicolons. For the complete font job, see the example under [Normal Font Data](#) on page 5-7.

```
?CONTROLBUF 103          ! FASTP Top Of Form Substitute
%33,"&a67r0c1L";        !      Posn to Row 67 Col 0 Left Margin 1
?CONTROLBUF 103          ! FASTP Top Of Form Substitute
%33,"&a1L",%14,%15;      !      Left Margin 1, FF CR
```

Example of User-Character Translation (CONTROLBUF 101)

The following example provides a translation table through a CONTROLBUF 101 directive to instruct FASTP to replace some nonprintable characters with spaces. This translation changes nonprintable characters that can produce incorrect output. The CONTROLBUF operation must be 101.

CONTROLBUF data contains the characters to be translated. Each translation is given as a pair of bytes. The first byte in each pair is translated into the second byte. There is no need to include characters that do not need translation.

```
?START-FONT
?FONT CTRLSPACE
?CONTROLBUF 101
! Input      Output
  %000      ,   %040      ! Null to space
  %003      ,   %040      ! %3    to Space
  %200      ,   %040 ;    ! %200 to space
?END-FONT
! Configure the #INVOICE location with the CTRLSPACE FONT
?START-LOCATIONS
! Location           Font           Device
#INVOICE.DEFAULT    CTRLSPACE    $LP2
! ...
```

Example of Top-of-Form Substitution (CONTROLBUF 103)

The following example shows a font job that can load alternative top-of-form sequences into the FASTP print process. The sequences remain in effect for the duration of the font job. The script file in this example creates font job LPX8D which allows 8 spooler pages per 8-1/2 by 11-inch sheet to print on a 5577 laser printer. Successive top-of-form sequences are replaced so that the printer is positioned to a new point on the current page instead of using a new page each time.

The CONTROLBUF operation must be 103. CONTROLBUF data contains the substitute top-of-form sequence. The multiple calls to CONTROLBUF 103 load several substitute sequences. FASTP will use each in turn.

```
?START-FONT
?FONT LPX8D      ! FONT to print 8 pages per 8 1/2 X 11 sheet
?SETMODE 6 0 1      ! Transparent
%33,"(8U",%33,"(s0p3.6v0s0b0T" ! Fixed line printer 3.6 Pt
%33,"&l10"          ! Landscape
%33,"&k4.78H"        ! HMI = 4.78 120ths
%33,"&l2.65C"        ! VMI = 2.65 48ths
%33,"&l1S"          ! Duplex long-edge binding
%33,"&a6L"          ! Left margin 6
?SETMODE 6 1 0      ! Normal
! This sets up the printer to do 4 Tandem pages per side
! by supplying a list of substitute top-of-form sequences
! to FASTP
?CONTROLBUF 103      ! Top-of-form substitute
```

```

!          Draw page dividers
%33,"&a67r0C"          !          Posn to row 67 col 0
%33,"*c3389a1b25g2P" !          Draw line 3389 dots x 1 dot
!                      !          x 25%
%33,"*p0Y",%33,"&a133C" ! Posn to dot 0 col 133
%33,"*c1a2338b30g2P" !          Draw Line 1 dot x 2338 dots x 30%
%33,"&a68r0c0L";          !          Posn to row 68 col 0
!                      !          left margin 0
?CONTROLBUF 103          ! Top-of-form substitute
%33,"&a0r134c134L";      !          Posn to row 0 col 134
!                      !          left margin 134
?CONTROLBUF 103          ! Top-of-form substitute
%33,"&a68r134c134L";    !          Posn to row 68 col 134
!                      !          left margin 134
?CONTROLBUF 103          ! Top-of-form substitute
%33,"&a0L",%14,%15;     !          Left margin 0, FF CR
?END-FONT
?START-LOCATIONS
#LISTING.DEFAULT        LPX8D        $LP2
! ...

```

Example of Virtual Font Jobs With FASTP (CONTROLBUF 113)

The FASTP print process allows sites to have an almost unlimited number of font jobs by allowing a single font job to contain many font sections, where any particular section applies only to certain locations. In theory, you could have a different font job (section) for each of the 4095 possible locations that can be configured within the spooler by using 4095 different CONTROLBUF 113 directives. In practice, you probably want to keep the font jobs smaller and, therefore, more efficient and manageable by having only a few sections in each font job. The maximum number of font sections is 255 multiplied by 4095, where 255 is the maximum of font jobs allowed by the spooler and each font job could have 4095 font sections.

Sectioning is achieved by using CONTROLBUF 113 directives embedded in a font job. The following example is a font script file that configures a single font job split into two sections with CONTROLBUF 113 directives. The first section applies to a single location. The second section applies to two different locations.

```

?START-FONT
?FONT COLORGROUP1

! This section applies only to #LASER.BLUE
! A ruler:          1234567890123456
?CONTROLBUF 113 "#LASER BLUE ";
"This is font job data for blue jobs"
"This is font job data for blue jobs"
"This is font job data for blue jobs"
"This is font job data for blue jobs"

! This section applies to #LASER.PINK and #LASER.RED
! A ruler:          1234567890123456
?CONTROLBUF 113 "#LASER PINK "

```



```

                "#LASER  RED      ";
"This is font job data for pink and red jobs"
"This is font job data for pink and red jobs"
"This is font job data for pink and red jobs"
"This is font job data for pink and red jobs"

?END-FONT

?START-LOCATIONS
! Location      Font           Device
#LASER.BLUE    COLORGROUP1  $LASER.#A
#LASER.PINK    COLORGROUP1  $LASER.#A
#LASER.RED     COLORGROUP1  $LASER.#A
?END-LOCATIONS

```

Note. In the CONTROLBUF 113 call, location names must be blank-filled to 16 characters in length. The destination need not be specified to match location groups. The location name is not case-sensitive. The data for the CONTROLBUF call is delimited by a semicolon (;).

SETMODE Directive

The SETMODE directive imbeds a SETMODE procedure call in the font job.

```
?SETMODE function param1 param2
```

function

is a device-dependent function. You must specify a number in the range 0 through 32767. See the SETMODE procedure in the *Guardian Procedure Calls Reference Manual* for the functions.

param1

is the parameter described with the SETMODE procedure in the *Guardian Procedure Calls Reference Manual*. You must specify a number in the range 0 through 32767.

param2

is the parameter described with the SETMODE procedure in the *Guardian Procedure Calls Reference Manual*. You must specify a number in the range 0 through 32767.

Example

```
?SETMODE 6 0 1           ! Enable transparent printing
```

SOURCE Directive

The SOURCE directive identifies a file containing data to be included in the font job. The Font utility copies the entire contents of the file except beginning or trailing bytes, as specified by SOURCE directive options.

```
?SOURCE source-file [ beginning-count [ trailing-count ] ]
```

source-file

identifies the file whose contents are copied as part of the data for the font job. This file must be an existing spooler job file (file code 129) or a file with a record size smaller than 900 bytes that can be read with the Guardian sequential I/O (SIO) routines. A SOURCE directive in a source file has no effect, so source files cannot be nested.

beginning-count

is the number of bytes to be skipped at the start of the first record in *source-file*. The value must be a number in the range 0 through 32767. At least one space must separate *beginning-count* from *source-file* and *trailing-count*.

trailing-count

is the number of bytes to be skipped at the end of the last record in *source-file*. The value must be a number in the range 0 through 32767.

Considerations

- The *beginning-count* and *trailing-count* parameters are integers.
- In counting bytes in the source file, count 1 byte for each character and 1 byte for each blank at the beginning of or within a line.
- Only data, as described under [Normal Font Data](#) on page 5-7, is allowed in the source file.
- The *beginning-count* and *trailing-count* parameters allow data created by various personal computer (PC) packages or workstation packages to be used as input for font jobs. Frequently, these packages include escape sequences that were intended to assist in printing but that prevent data from being used in certain font jobs. These offending sequences usually occur at the beginning or end of the data. For example, the data might start with a PCL Esc E, which is valid if used in a PCL macro command, but which is invalid in other cases. The data might end with a form feed, which could cause an extra blank page to be printed. The *beginning-count* and *trailing-count* options lets you skip over these offending sequences.

Example

```
?SOURCE INVOICE 22 3      ! Copy data from the file INVOICE and
                          ! trim the first 22 and last 3 bytes.
```

START-LOCATIONS and END-LOCATIONS Directives

The START-LOCATIONS directive indicates the start of the script file locations section, which describes locations to which the font should be attached and the devices to which the locations should be linked. Location entries identify the locations, fonts, and devices. The END-LOCATIONS directive indicates the end of the locations section.

```
?START-LOCATIONS
  { location-name [,] font-name [,] [ device-name ] } ...
?END-LOCATIONS
```

location-name

is the logical destination of a job. If a print device is associated with a specific location, that print device becomes the physical destination of the job. The location is a two-part name: *#group.dest*.

The *group* parameter is 1 through 7 alphanumeric characters starting with a letter. The # character must always precede the *group* designation.

The *dest* parameter is 1 through 8 alphanumeric characters starting with a letter.

If the location name does not currently exist in the spooler, a new location with this name is created. If the location name currently exists in the spooler, those font and device values are changed to the values specified in these directives.

font-name

is the name of an existing font, that could have been defined earlier in the script file or could have been previously added to the spooler font list.

device-name

is the name of a spooler device to be attached to this location. If no device is specified, the added location is not linked to a device. If no device is specified and the location currently exists in the spooler, then the device currently specified for that location remains the same.

Example

```
?START-LOCATIONS
#LOCAL.P1      LOGOFONT      $LPTR.#A
#REMOTE.PTR ,  TESTFONT ,    \NODE.$LASER.#X
#EXISTING.A    TESTFONT
?END-LOCATIONS
```

Comments in the Script File

Comments can appear anywhere in the script file and are ignored by the Font utility.

A comment is

- Any text before the first START-FONT directive
- Any text to the right of an exclamation point (!)
- Any text between the END-FONT and the START-LOCATIONS directives
- Between the START-LOCATIONS and END-LOCATIONS directives, any line that does not start with the # sign
- Any text following an END-LOCATIONS directive

Examples

The following example demonstrates comments. The lines are numbered for demonstration; the line numbers are not part of the test. Comment lines are lines 1, 2, 8, 9, 11, 12, 14, 16, 17, 19, and 20. Comments follow the exclamation points (!) in lines 5 and 6.

1. Beginning text lines are comments
2. and are ignored.
3. ?START-FONT
4. ?FONT DEMO
5. ?CONTROL 1 0 ! Issue a form feed
6. %33,"E" ! Issue a PCL reset
7. ?END-FONT
8. These lines are treated as comments and
9. are ignored. They can contain anything.
10. ?START-LOCATIONS
11. Comments comments comments
12. Comments comments comments
13. #RETAIL.DEFAULT RETAILFORM \$LASER.#A
14. Comments comments comments
15. #TRADE.DEFAULT TRADEFORM \$LASER.#B
16. Comments comments comments
17. Comments comments comments

18. ?END-LOCATIONS
19. Comments comments comments
20. Comments comments comments

Script File Example

The following is an example of a script file. This file is input to the Font utility, which loads font jobs in the spooling system. These font jobs are used by locations with Laser-LX and other similar PCL devices attached. The script file also supplies information to enable the configuration of these locations.

Initial text is comments.

```
?START-FONT      ! Identifies the start of the FONT section
                  ! that describes font jobs.
?FONT PORTRAIT   ! An example PCL font
! Comments go here
?SETMODE 6 0 1   ! Transparent ON
%33,"E"          ! Reset
%33,"&l00"       ! Portrait
%33,"&l6D"       ! 6 lines per inch
%33,"&l1S"       ! Duplex long-edge binding
%33,"(8U"        ! Roman 8
%33,"(s12V"     ! Point 12
%33,"(s10H"     ! Pitch 10
?SETMODE 6 1 0   ! Transparent OFF
?CONTROL 1 0     ! Form feed

?FONT TRANSP     ! Another example font job, this
                  ! one enables transparent printing
?SETMODE 6 0 1   ! Transparent ON
?END-FONT        ! Identifies the end of the section.
! This completes the FONT section.

?START-LOCATIONS ! Identifies the LOCATIONS section.
                  ! Location entries describe
                  ! locations to be added

! Note that the format of location entries is:
!   <location> [,] <font-name> [,] <device>
!   #LX.PORT          PORTRAIT          $LASER.#A
!   #LX.GLASS         TRANSP            $LASER.#A
?END-LOCATIONS   ! Identifies the end of LOCATIONS
section.
```

Notice that the file has two sections: The first describes the font jobs to be created, and the second details the locations.

A directive has a question mark (?) as the first nonblank character.

Text preceding the START-FONT directive is treated as a comment. Text to the right of an exclamation point (!) is treated as a comment.

6 RPSetup Utility

The RPSetup utility establishes cross-spooler connections between spoolers in a network. These connections give users access to printers on remote nodes in a network. You use RPSetup to create and maintain a spooler configuration for a node with a limited number of printers, so that print jobs from local applications can be sent to a remote spooler and queued for printing on remote printers.

The following topics are described in this section:

- [How the RPSetup Utility Works](#) on page 6-2
- [Running RPSetup](#) on page 6-6
- [Configuring Entities in the Local Spooler](#) on page 6-8

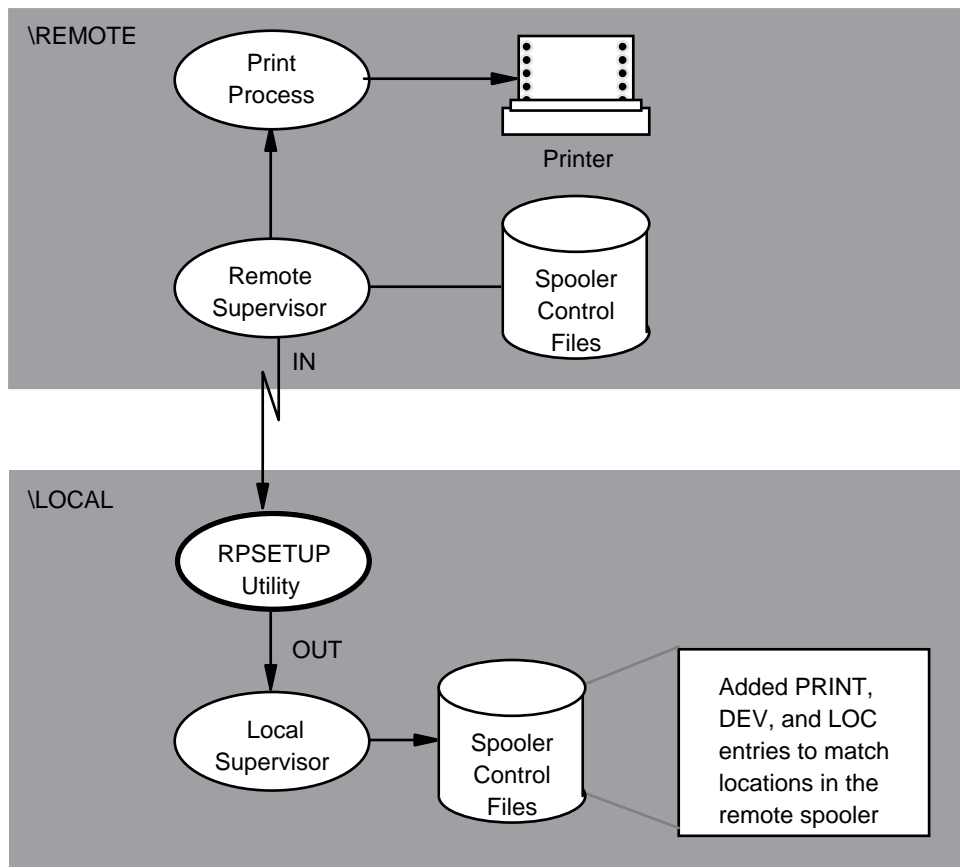
How the RPSetup Utility Works

When RPSetup runs, it reads configuration information from the remote spooler supervisor, analyses the information, and writes new configuration information to the local spooler supervisor.

RPSetup sets up connections to all valid locations on the remote node. To create these connections, RPSetup creates locations on the local spooler with the same names as locations on the remote node. RPSetup might also add required print processes and devices to the local spooler configuration.

The functions of the RPSetup utility are illustrated in [Figure 6-1](#).

Figure 6-1. Configuration Functions of the RPSetup Utility



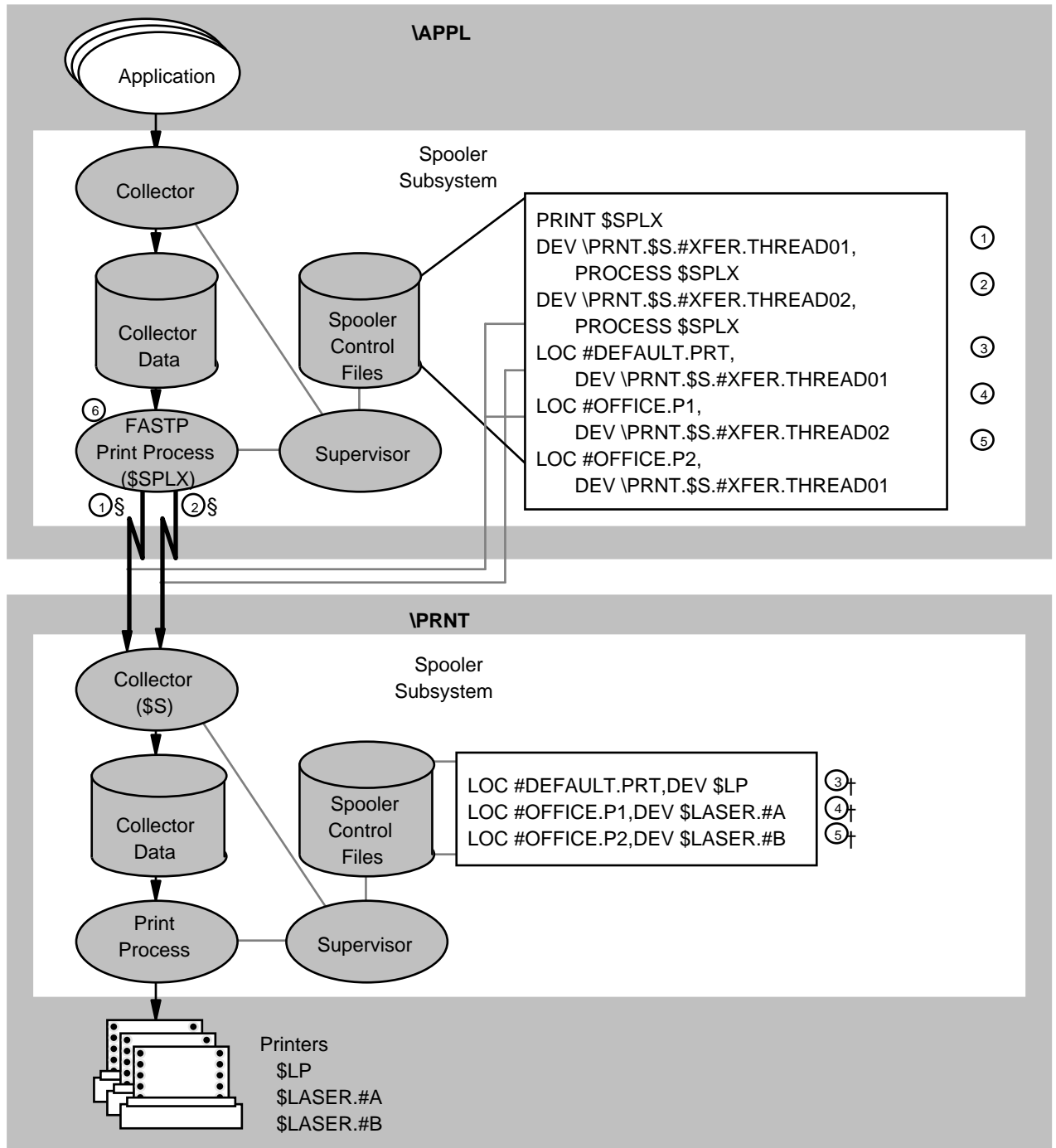
CDT 014CDD

Interspooled Threads

In creating cross-spooled connections, RPSetup creates **interspooled threads** to provide spooler-to-spooled transfers for print jobs. An interspooled thread is a spooled device that is configured as a collector for a different spooled. An interspooled thread is a path from one spooled to another. The number of threads equals the number of print jobs that can be simultaneously transferred to the other spooled.

Two interspooled threads previously established by RPSetup are illustrated in [Figure 6-2](#), which shows the locations on the remote node \PRNT configured as locations associated with threads on the local node \APPL.

Figure 6-2. Interspooled Threads



CDT 015CDD

The numbers in [Figure 6-2](#) mark the following elements:

- 1,2 Mark the definition of the interspooler threads. Each is a device configured as a collector (\PRNT.\$S ...) and uses print process \$SPLX (which is a process name for the FASTP program) to control the interspooler thread.
- 3,4,5 Mark the definition of particular locations with cross-connections to remote node \PRNT. These locations, rather than being associated with a local printer, are associated with an interspooler thread to transfer jobs to \PRNT.\$S for subsequent printing on printers associated with those particular location names.
- 6 Marks the print process managing the interspooler threads. When print jobs are sent to the locations associated with interspooler threads, FASTP transfers the jobs as directed by the configuration to a collector on the remote node.
- 1\$,2\$ Mark the interspooler threads. Each is a path for print jobs from one spooler to another.
- 3+,4+,5+ Mark the definition of remote (target) locations for print jobs sent through the interspooler threads. For example, a print job sent to LOC #OFFICE.P2 on \APPL is transferred to the spooler collector on \PRNT to LOC #OFFICE.P2 for printing on the associated printer \$LASER.#B.

When setting up the interspooler threads to cross-connect the two spoolers, RPSetup issues the following kinds of commands to change the spooler configuration. Examples of these commands are shown in the spooler control files on node \APPL in [Figure 6-2](#).

- PRINT Print process \$SPLX (default process name for the FASTP print process), which controls spooler-to-spooler transfers through the interspooler threads. FASTP is the most efficient print process supplied by the spooler subsystem for this task.
- DEV Device configured as a collector, which is an interspooler thread.
- LOC Location associating a remote printer with an interspooler thread.

Running RPSetup

To run RPSetup, enter the following command at the TACL prompt:

```
RPSETUP / IN remote-spooler , OUT local-spooler /
```

remote-spooler

is the supervisor process name of a remote spooler configured with printers to which remote connections are to be established. To specify *remote-spooler*, use the network form of the supervisor process name (for example, \FAR.\$SPLS).

local-spooler

is the supervisor process name of a local spooler whose configuration is to have locations and devices added to provide locations for actual printing through *remote-spooler*. RPSetup and *local-spooler* should execute on the same node.

Considerations

- RPSetup displays any status or error information on the terminal where the RUN command is used.
- Error messages are described in [Appendix E, RPSetup Utility Messages](#).
- Run the RPSetup utility whenever the remote spooler has device or print process changes. Specify *remote-spooler* and *local-spooler* in the command to run RPSetup, and the utility updates the local spooler configuration.
- RPSetup establishes interspooler threads and configures printers and devices as specified by PARAM and ASSIGN commands associated with the RPSetup utility. Unless you set these values with specific commands, default values apply as described under [Configuring Entities in the Local Spooler](#) on page 6-8.
- RPSetup sets up connections on the local spooler to all qualifying locations on the remote system. A qualifying location is one linked to a device that is not a collector.

In setting up the connections, RPSetup assigns an interspooler thread to each qualifying location, giving the first location thread 01, the second location thread 02, and so on until each configured interspooler thread is used; and it repeats that assignment process until all locations are assigned.

- FASTP is the default print process for RPSetup configurations; therefore, RPSetup configures interspooler threads by using values appropriate for FASTP operations. The ASSIGN PRINTFILE command can be used to change the print process.
- The FASTP print process, if used to control interspooler threads for spooler-to-spooler job transfer, can also be assigned to other devices on *local-spooler*.

- If you use a print process other than FASTP, you must ensure that the default values of the configuration parameters are appropriate for the print process you use. Because the default configuration parameters are set for the FASTP print process, you might need to specify PARAM commands to set configuration parameters specifically for the print process you use. For some default operations performed by FASTP, see [PARAM DEVPARM](#) on page 6-13. For all the configuration parameters of print processes, see the DEV command.
- Default values for RPSetup are set with the assumption that the FASTP print process retains the source job location and job ownership when the job is copied to the target spooler, but that FASTP omits any HOLDAFTER flag for a transferred print job. For information about controlling FASTP default operation, see [PARAM DEVPARM](#) on page 6-13.
- Setting up more than one thread between two spooler systems provides more efficient use of any Expand connection than a single thread would. The number of threads equals the number of print jobs that can be simultaneously transferred to a remote spooler. The [PARAM NUMTHREADS](#) command controls the number of interspooler threads configured (the default number is 2).

Requirements for Running RPSetup

These are the requirements for running the RPSetup utility:

- You must either be a system operator (user ID 255, *n*) or be the owner of the local spooler processes.
- You must have a remote password and be able to logon to the remote system.
- If *remote-spooler* is protected by the Safeguard security facility, your user ID must meet Safeguard requirements to access *remote-spooler*.
- Before issuing the RPSetup command, you must issue ASSIGN and PARAM commands for configuration parameters whose default values are not appropriate for your spooler.
- You must run the RPSetup utility on the local node to access and change spooler configuration files.
- The remote spooler must have at least one collector in the active state.
- *remote-spooler* and *local-spooler* are usually on different nodes of an Expand network, with one rare exception. Occasionally, both spoolers will be on the same node when there are more devices than a single spooler can support.

Example

Assume that the \PRNT.\$SPLS spooler has the following locations:

```
COLLECT $S
LOC #DEFAULT.PRT,DEV $LP
LOC #OFFICE.P1,DEV $LASER.#A
LOC #OFFICE.P2,DEV $LASER.#B
```

The following command to run RPSetup sets up the spooler on \APPN so that all printing locations on \PRNT are also available on the \APPN spooler.

```
RUN RPSETUP /IN \PRNT.$SPLS, OUT \APPN.$SPLS /
```

RPSetup installs the devices and locations listed below in \APPN.\$SPLS; this example is illustrated in [Figure 6-2](#), earlier in this section.

```
PRINT $SPLX
DEV \PRNT.$S.#XFER.THREAD01,PROCESS $SPLX
DEV \PRNT.$S.#XFER.THREAD02,PROCESS $SPLX
LOC #DEFAULT.PRT,DEV \PRNT.$S.#XFER.THREAD01
LOC #OFFICE.P1,DEV \PRNT.$S.#XFER.THREAD02
LOC #OFFICE.P2,DEV \PRNT.$S.#XFER.THREAD01
```

After the connections are established by RPSetup, an application on \APPN can spool to the local \$S.#DEFAULT location and reach \$LP on \PRNT.

Configuring Entities in the Local Spooler

The RPSetup utility configures entities in the local spooler. All configuration parameters have reasonable default values, but you can change the default configuration by including the ASSIGN and PARAM commands before you run the RPSetup utility.

Default Configuration

The default configuration is summarized in [Table 6-1](#).

Table 6-1. ASSIGN and PARAM Command Default Values (page 1 of 3)

RPSetup Parameter	Description	Default Value	Corresponding SPOOLCOM Command
ASSIGN PRINTFILE	Establishes print process	\$SYSTEM.SYSTEM.FAST P	None
PARAM BUPCPU	Sets backup processor for the print process	Backup processor for local supervisor	PRINT BACKUP
PARAM DEVEXCLUSIVE	Sets ownership mode of the device	OFF (print process should close the device when no more jobs are waiting)	DEV EXCLUSIVE

Table 6-1. ASSIGN and PARAM Command Default Values (page 2 of 3)

RPSetup Parameter	Description	Default Value	Corresponding SPOOLCOM Command
PARAM DEVFIFO	Sets algorithm by which jobs are selected for printing on the device	OFF (not queued in FIFO)	DEV FIFO
PARAM DEVHEADER	Sets header to precede each job	Ignored by FASTP for interspooler job transfers; can be set for individual devices	DEV HEADER
PARAM DEVPARM	Sets a device parameter passed to the print process	0	DEV PARM
PARAM DEVRESTART	Sets the interval the device waits between automatic restart attempts	30 seconds	DEV RESTART
PARAM DEVTRUNC	Specifies whether lines longer than the device width are truncated or wrapped around	OFF (sets wrap around)	DEV TRUNC
PARAM DEVWIDTH	Sets maximum line length for the device	-1 (specifies obtaining the record size from DEVICEINFO procedure)	DEV WIDTH
PARAM NUMTHREADS	Establishes the number of interspooler threads	2	None
PARAM PRICPU	Sets the processor in which the print process that controls the interspooler threads executes	Processor for local supervisor	PRINT CPU
PARAM PRINTNAME	Is the name of the print process	Checks ASSIGN PRINTFILE for agreement with print process program (default process name is \$SPLX)	PRINT <i>\$processname</i>

Table 6-1. ASSIGN and PARAM Command Default Values (page 3 of 3)

RPS Setup Parameter	Description	Default Value	Corresponding SPOOLCOM Command
PARAM PRINTPARAM	Sets parameters	0	PRINT PARM
PARAM PRIORITY	Sets the priority of the print process that controls the interspooler threads	Target supervisor priority minus 10	PRINT PRI
PARAM THREADNAME	Sets the generic portion of the device name for interspooler threads	Default name of the first active collector in IN <i>remote-spooler</i> with this qualifier: #XFER.THREADnn	None

ASSIGN and PARAM Commands

The syntax for the commands listed in [Table 6-1](#) is described in the following subsections.

ASSIGN PRINTFILE

The ASSIGN PRINTFILE command specifies the Guardian file name of the program file for the print process controlling the interspooler thread. This configuration parameter applies individually to all interspooler threads.

```
ASSIGN PRINTFILE , object-filename
```

object-filename

is the object file name, in local format, of the print process. The default name is \$SYSTEM.SYSTEM.FASTP.

PARAM BUPCPU

The PARAM BUPCPU command configures the print process that controls the interspooler threads. The command specifies the backup processor and is equivalent to the Spoolcom command PRINT BACKUP.

```
PARAM BUPCPU backup-cpu
```

backup-cpu

is an integer in the range 0 through 15 specifying the backup processor for the print process that controls the interspooler threads. The default processor is the backup processor for the local spooler supervisor.

PARAM DEVEXCLUSIVE

The PARAM DEVEXCLUSIVE command specifies the ownership mode of the device. This configuration parameter applies individually to all interspooler threads. The command is equivalent to the SPOOLCOM command DEV EXCLUSIVE.

```
PARAM DEVEXCLUSIVE { ON | OFF [ ! ] }
```

ON

specifies that the print process should not close the device when the printer is not printing a job.

OFF

specifies that the print process should close the device when no more jobs are waiting to be printed. The default value is OFF.

OFF!

specifies that the device is to be closed between jobs.

PARAM DEVFIFO

The PARAM DEVFIFO command specifies the algorithm by which jobs are selected for printing on this device. This configuration parameter applies individually to all interspooler threads. The command is equivalent to the SPOOLCOM command DEV FIFO.

```
PARAM DEVFIFO { ON | OFF }
```

ON

specifies first-in, first-out (FIFO) queuing.

OFF

specifies that jobs are not queued in FIFO order but are queued according to an algorithm that takes into account the relative length of an incoming job and how long the other jobs have been in the queue. The default value is OFF.

PARAM DEVHEADER

The PARAM DEVHEADER command specifies whether a standard or batch header page should precede each job. This configuration parameter applies to interspooler threads controlled by a print process other than FASTP. This parameter is ignored by the FASTP print process when a controlling job transfers between spoolers. These jobs are always transferred without additional header or trailer pages.

At the target node, the print process can choose to add these header or trailer pages according to settings configured for the real device. The PARAM DEVHEADER command is equivalent to the SPOOLCOM command DEV HEADER.

```
PARAM DEVHEADER { ON | OFF | BATCH }
```

ON

specifies that a header page should print at the beginning of every job.

OFF

Specifies that header pages are not to be printed at the beginning of jobs. The default value is OFF.

BATCH

specifies that two header pages and three trailer pages (also containing job information) print for every job. This enables jobs printed on accordion-fold paper to have a header page appearing on top regardless of whether the job begins on

an odd or even page. Printing over the page folds on the trailer pages enables jobs to be easily separated from each other.

PARAM DEVPARM

The PARAM DEVPARM command specifies a device parameter that is passed to the print process controlling this device whenever the supervisor communicates with the print process. This configuration parameter applies individually to all interspooler threads. The command is equivalent to the SPOOLCOM command DEV PARM.

```
PARAM DEVPARM parameter
```

parameter

is a number that can be represented in 16 bits. The range for *parameter* is -32768 through +32767. The default value is 0, which specifies that the print process retain the source job location and job ownership when the job is copied to the target spooler. Other default attributes are described in [Table 6-2](#).

The default value of PARM DEVPARM is set for the FASTP print process. If you use another print process, you must ensure that the value is appropriate for that print process.

This parameter allows you to control operations performed by the print process when transferring print jobs from the local (source) spooler to a remote (target) spooler. Some job attributes that FASTP sets for a print job delivered to a target spooler are listed in [Table 6-2](#) together with the bit values to change the operation.

Table 6-2. Default Attributes Set by FASTP on Interspooler Transferred Print Jobs (page 1 of 2)

Bit	Value	Attribute of Transferred Print Jobs	FASTP's Default Operation
< 10 >	0	Retains the same job ownership at the target spooler as at the source spooler	Equals 0. For example, if user ADMIN.OPERATOR submitted a job at the source node, FASTP sets ADMIN.OPERATOR as the job's owner at the target node.
	1	Sets job ownership (PROCESSACCESSID) at target node to the print process performing the transfer	
< 11 >	0	Omits any HOLDAFTER flag for a job transferred to target spooler	Equals 0. FASTP delivers all jobs without a HOLDAFTER flag to the target spooler.
	1	Always sets the HOLDAFTER flag for a job transferred to a target spooler	

Table 6-2. Default Attributes Set by FASTP on Interspooled Transferred Print Jobs (page 2 of 2)

Bit	Value	Attribute of Transferred Print Jobs	FASTP's Default Operation
< 12 >	0	Omits any HOLDAFTER flag for a job transferred to target spooler	Same as < 11 >.
	1	Retains any HOLDAFTER flag for a job transferred to a target spooler	
< 13 >	0	Sets target location name to that of the source job location	Equals 0. For example, a job sent to location \$S.#OFFICE.A has this same location name after being transferred to the target spooler.
	1	Sets target location name to that of the device thread name	

For more information on parameter values, see the DEV PARM command in Section 4, Spoolcom.

PARAM DEVRESTART

The PARAM DEVRESTART command sets the interval the device waits between automatic restart attempts. This configuration parameter applies individually to all interspooled threads. The command is similar to the SPOOLCOM command DEV RESTART.

```
PARAM DEVRESTART interval
```

interval

is the number of seconds the device waits between automatic restart attempts. The value is in the range 10 through 32767 seconds. The default is 30 seconds.

PARAM DEVTRUNC

The PARAM DEVTRUNC command specifies whether lines longer than the device width are to be truncated or wrapped around. This configuration parameter applies individually to all interspooled threads. The command is equivalent to the SPOOLCOM command DEV TRUNC.

```
PARAM DEVTRUNC { ON | OFF }
```

ON

causes the print process to discard the ends of spooled lines that are longer than the device width.

OFF

causes the print process to print on the next sequential line the ends of any spooled lines that are longer than the device width. The default value is OFF.

PARAM DEWIDTH

The PARAM DEWIDTH command configures the maximum line length for the device. This configuration parameter applies individually to all interspooler threads. The command is equivalent to the SPOOLCOM command DEV WIDTH.

```
PARAM DEWIDTH device-width
```

device-width

specifies the maximum line length for the device. The range of values is 0 through 32676; -1 is also valid. A value of -1 (the default value) causes the print process for the device to obtain the record size from the file-system DEVICEINFO procedure. A value of 0 indicates that there is no limit to width.

PARAM NUMTHREADS

The PARAM NUMTHREADS command establishes the number of interspooler threads. Each thread is a device (DEV entry) in the local spooler configuration. The number of threads establishes the maximum number of print jobs that can simultaneously be transferred from the local spooler to the remote spooler.

```
PARAM NUMTHREADS number
```

number

is the number of threads. The value can be 1 through the maximum number of devices supported by the assigned print file ([ASSIGN PRINTFILE](#) command). For FASTP—the spooler-released print processes—the maximum values possible is 32. The default value is 2.

PARAM PRICPU

The PARAM PRICPU command sets the processor in which the print process that controls the interspooler threads executes. This command is equivalent to the SPOOLCOM command PRINT CPU.

```
PARAM PRICPU cpu
```

cpu

is a number in the range 0 through 15 specifying the processor where the print process executes. The default processor is the processor in which the local spooler supervisor executes.

PARAM PRINTNAME

The PARAM PRINTNAME command sets the process name of the print process that controls the interspooler thread. This configuration parameter applies individually to all interspooler threads. This command is equivalent to the SPOOLCOM command PRINT *\$process-name*.

```
PARAM PRINTNAME $process-name
```

\$process-name

is the process name (in local form) of the print process. If the name is already configured in the local spooler, the name is used if the print program it represents successfully matches the print program represented by the object file name (from the [ASSIGN PRINTFILE](#) command). The object file and process name must represent the same print program. If the object file name does not match and is inactive, RPSetup changes the object file to match. The default process name is \$SPLX.

PARAM PRINTPARAM

The PARAM PRINTPARAM command sets a print-process parameter that is passed by the supervisor to the print process in the Startup message. Its meaning is defined by a user-written print process. The Tandem print process, FASTP, does not use this parameter. This command is equivalent to the SPOOLCOM command PRINT PARM.

```
PARAM PRINTPARAM parameter
```

parameter

is a number that can be represented in 16 bits. The range for *parameter* is -32768 through +32767. The default value is 0.

PARAM PRIORITY

The PARAM PRIORITY command sets the priority of the print process that controls the interspooler threads. This command is equivalent to the SPOOLCOM command PRINT PRI.

```
PARAM PRIORITY execution-priority
```

execution-priority

is the execution priority. The value can be in the range 1 through 199. The default value is the priority of the local spooler supervisor minus 10.

PARAM THREADNAME

The PARAM THREADNAME command sets the generic portion of the device name for an interspooler thread. This configuration parameter applies individually to all interspooler threads. You can use the command to override the entire DEV name, specified through the Spoolcom DEV command, including the remote collector name. You can also use this parameter to override the *#qualifier.name* portion only.

```
PARAM THREADNAME name
```

name

is a device name. The default name is the first active collector in the IN *remote-spooler* parameter of the RPSetup command with the following added: #XFER.THREAD*nn*, where *nn* is a number in the range 01 through 99 assigned by RPSetup to indicate the ordinal number of the thread.

A SPOOL Program

The SPOOL program is used to run the spooler supervisor as part of coldstarting and warmstarting the spooler. This appendix describes the following topics:

- [Running the SPOOL Program](#) on page A-1
- [EMS Collector DEFINE](#) on page A-4

Running the SPOOL Program

You can run the SPOOL program by entering the following command at the TACL prompt:

```
SPOOL /IN control-filename[.OUT [ log-file ] ]  
      , NAME $supervisor-process  
      [, PRI execution-priority ]  
      [, CPU primary-cpu ]  
      / [ backup-cpu [ ,coldstart-parameters ] |  
      R[EBUILD] [, backup-cpu ] ,coldstart-parameters ]
```

IN *control-filename*

specifies a file that contains the names and attributes of the processes, locations, and devices that make up the system. The file name *control-filename* has the same form as a disk file name except that you specify at most seven letters or digits; the supervisor appends a single digit (0-9) to the specified *control-filename* to create a unique file name. See [Considerations](#) on page A-3 for more information.

OUT [*log-file*]

is the name of the device, process, or disk file where the spooler logs messages.

If this parameter is omitted or if for some reason the supervisor cannot communicate with the specified log file, messages are logged on the system console (\$0). For example, if the log file is a disk file that does not exist or was not created, messages are logged to the system console.

If OUT is present but *log-file* is omitted, \$0 is assumed to be the log file.

NAME *\$supervisor-process*

is the process name of the supervisor. The name \$SPLS is recommended but not mandatory.

PRI *execution-priority*

is the execution priority of the supervisor. The range is 1 through 199. Specifying a value greater than 199 causes the supervisor to run at priority 199.

The default priority is 1 less than the priority of the command interpreter from which the supervisor is run.

CPU primary-cpu

is the processor number of the processor module (0 through 15) in which the supervisor's primary process runs.

The default processor module is the same processor module that the command interpreter is located in.

backup-cpu

is the processor number of the processor module (0 through 15) in which the supervisor's backup process is to run.

If this parameter is omitted, the supervisor runs without a backup.

coldstart-parameters

are the parameters described below. The first five of these parameters are required when the spooler is coldstarted.

You must also include the REBUILD option in order to increase any of the parameters in a warm start. You cannot decrease any values of *coldstart-parameters* without cold starting the spooler.

If you do not use the REBUILD option, *coldstart-parameters* cannot be changed without cold starting the spooler again.

num-of-jobs

is the maximum number of jobs that the spooler will handle at any one time. The range is 2 through 8191. There is no default.

num-of-locations

is the maximum number of destinations and groups the routing structure will contain. For example, the number of locations for a routing structure containing only the locations #DEFAULT.DEFAULT and #MYGRP.YRDEST must be at least four: two locations plus two groups. The range is 2 through 4095. There is no default.

num-of-devices

is the maximum number of devices that can be known to the spooler at any one time. The range is 1 through 4095 for D48, D46, and D41 spooler. There is no default.

num-of-collectors

is the maximum number of collectors that will ever be declared for this spooler. The range is 1 through 16. There is no default.

num-of-print-processes

is the maximum number of print processes that will be declared for this spooler. The range is 1 through 45. There is no default.

num-of-fonts

is the maximum number of font jobs that will ever be declared for this spooler. The range is 1 through 255. The default is 16.

num-of-batch-jobs

is the maximum number of batch jobs that will ever be declared for this spooler. The range is 1 through 4095. The default is 256.

num-of-device-queue-nodes

is the maximum number of device queue nodes. Every job needs a minimum of 1 device queue node. The optimum value of this parameter is twice the number of jobs. The default value is 131086.

R [EBUILD]

is used during a warmstart to specify new coldstart parameters or to recover from control-file corruption. *cold-start-parameters* must be specified when you use this option. See [Considerations](#) for more information.

Considerations

- The supervisor forms the names of its control files by appending digits from 0 through 9 to the specified *control-filename*.
- When cold starting a spooler, make sure that none of the *control-filename* files exist, because the supervisor attempts to create them. If a file already exists, the supervisor will fail and terminate. The spooler might not use all of these files, but you should keep them reserved for use by the spooler.
- You must specify NAME *\$supervisor-process*, because the spooler will not run an unnamed process. It is recommended that you use *\$SPLS* as the name for the spooler supervisor. Running SPOOLCOM without specifying a particular supervisor causes SPOOLCOM to communicate with a process named *\$SPLS*.
- The process name assigned to a supervisor on a warmstart need not be the same as the process name assigned to the supervisor the last time the spooler was made active. It is the control file name rather than the supervisor process name that uniquely identifies a particular spooler.
- The REBUILD option causes the supervisor to rebuild the spooler control files. One symptom of a corrupted control file is a corrupted job map. This can be seen if you warmstart a spooler and find a listing for a job 0 when you execute the JOB command (indicating that the new spooler has a corrupted job map).

- If a recovery from control-file corruption is to be done during a SPOOL REBUILD operation, the coldstart parameters specified in the command string must be the same as those values that were specified when the spooler was coldstarted.

Examples

1. To create the spooler supervisor process, specifying a maximum of 4095 jobs and 4095 locations and no more than 128 devices, 10 collectors, 10 print processes, 12 fonts, and 500 batch jobs, and 10,000 device queue nodes, enter:

```
6> SPOOL / IN $MKT.SPL.SPL, OUT $0, NAME $SPLS, NOWAIT, &
7> PRI 147, CPU 0/1, 4095, 4095, 128, 10, 10, 12, 500, 10000
```

2. To recover from a corrupted spooler control file, enter

```
8> SPOOL / IN $MKT.SPL.SPL, OUT $0, NAME $SPLS, NOWAIT, &
9> PRI 147, CPU 0/REBUILD, 4, 4095, 4095, 128, 10,& 10> 10,
12, 500
```

3. To warmstart a spooler, retaining the original specifications for maximum coldstart parameters except to increase the number of devices to the maximum (255), enter

```
11> SPOOL / IN $MKT.SPL.SPL, OUT $0, NAME $SPLS, NOWAIT, &
12> &PRI 147, CPU 0/REBUILD, 4, 4095, 4095, 255,&
13> 10,10,12, 500
```

EMS Collector DEFINE

The destination process name for the EMS event messages from all spooler components is established with a DEFINE provided by the process that runs SPOOL. The define is established for a particular EMS collector process with the following command:

```
ADD DEFINE =_EMS_COLLECTOR, FILE $process-name
```

If no define exists, the event messages will be directed to \$0.

B Peruse Messages

This appendix describes the Peruse error messages that are displayed on your terminal. For additional information on Peruse, see [Section 3, Peruse](#).

```
ACCESS TO SPOOL DATA FILE FAILED, FILE ERROR number
```

Cause. A file-system error occurred during input/output (I/O) to a spooler data file.

Effect. Peruse cannot access the data file for this job.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
ACCESS TO SPOOL SUPERVISOR PROCESS FAILED, FILE ERROR number
```

Cause. A file-system error occurred during input/output (I/O) to a supervisor process.

Effect. The operation terminates.

Recovery. Check the status of the supervisor using the Spoolcom SPOOLER command, or call the system operator. Also, refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for the file-system error *number*.

```
CAN'T CREATE 'OUT' FILE: number
```

Cause. The OUT file of a LIST command was a nonexistent disk file. Peruse tried to create the file and CREATE failed. *number* specifies the file-system error.

Effect. The operation terminates.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
CAN'T OPEN 'OUT' FILE: number
```

Cause. The Guardian OPEN procedure returned an error when Peruse tried to open the OUT file.

Effect. Peruse does not open the OUT file.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
CAN'T PURGE 'OUT' FILE: number
```

Cause. An exclamation point (!) was specified with the OUT file of a LIST command and the file could not be purged. *number* specifies the file-system error number.

Effect. The operation terminates.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
CAN'T SPOOLSTART 'OUT' FILE: number
```

Cause. SPOOLSTART failed for the OUT file of a LIST command. *number* specifies the spooler error number.

Effect. The operation terminates.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
COMMAND CANNOT BE ACCOMPLISHED GIVEN CURRENT SPOOLER STATE
```

Cause. The spooler is in the drain or dormant state, and the command required an operation that conflicted with the spooler state.

Effect. The command is ignored.

Recovery. Warmstart the spooler so that you can set and modify its attributes.

```
COMMAND NOT SUPPORTED FOR SPOOLER JOB FILES
```

Cause. A command was invoked that is not supported for spooler job files, and a spooler job file was the current job.

Effect. The command is not executed.

Recovery. If you intended to execute the command against a spooler job, make it the current job and retry the command.

```
CURRENT JOB HAS BEEN DELETED FROM SPOOLER
```

Cause. The command referred to a job that is no longer in the spooler.

Effect. The spooler ignores the command.

Recovery. Respool your job, and then reexecute your command.

```
DATA FILE OPEN FAILED, ERROR: number
```

Cause. The Guardian OPEN procedure returned file-system error *number* when you attempted to open a data file for the current job.

Effect. Peruse does not open the data file.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
EDITREAD ERROR: number
```

Cause. The Guardian EDITREAD procedure returned file-system error *number* when it attempted to read a command file that is in the EDIT format.

Effect. Peruse cannot read the command file.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
EDITREADINIT ERROR: number
```

Cause. The Guardian EDITREADINIT procedure returned file-system error *number* when it attempted to read a command file that is in EDIT format.

Effect. Peruse cannot use the command file.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
ENTRY IN USE
```

Cause. An attempt was made to delete a job that had not finished spooling. Peruse shows the job state as open.

Effect. The job is not deleted.

Recovery. Place the job on hold, if necessary, so it will not print when spooling is complete. Later, return to Peruse, remove the hold, and delete the job. If the job never finishes spooling, contact your system operator for assistance in deleting the job.

ENTRY NOT FOUND

Cause. The DEV command specified a device that is not defined to the spooler.

Effect. The spooler does not execute the command.

Recovery. Verify the correct device name (you can use the Spoolcom DEV command to list all devices on the system), and then reexecute your command.

FILE DOES NOT EXIST

Cause. You attempted to declare a nonexistent file as the current job.

Effect. The command is not executed.

Recovery. Declare an existing spooler job file as the current job.

FILE IS NOT SPOOLER JOB FILE

Cause. You attempted to declare a file that was not a spooler job file as the current job.

Effect. The command is not executed.

Recovery. Declare a spooler job file as the current job.

ILLEGAL COMMAND

Cause. A line was entered that did not start with a legal command.

Effect. The spooler does not execute that line.

Recovery. Check your syntax, and then reenter a valid command.

ILLEGAL OUT FILE SPECIFICATION

Cause. A command was entered that specified an illegal OUT file parameter.

Effect. The spooler does not execute your command.

Recovery. Check your syntax, and then reenter a command with the correct OUT file parameter.

ILLEGAL PAGE NUMBER

Cause. A PAGE command was entered with an illegal page-number parameter.

Effect. The spooler does not execute this command.

Recovery. Check your syntax, then reenter this command with the correct page number.

ILLEGAL PARAMETER

Cause. A command with an illegal parameter value was entered.

Effect. The spooler does not execute your command.

Recovery. Check your syntax, and then reenter this command with the correct parameter value.

ILLEGAL SUPERVISOR PROCESS NAME

Cause. A Peruse or Peruse OPEN command specified either no supervisor name or one that is not a valid process name.

Effect. You cannot obtain information about spooler jobs unless Peruse establishes communication with a supervisor process.

Recovery. Check your syntax, then reenter your command with the correct supervisor process name in remote or local form.

INPUT FILE ERROR: *number*

Cause. During an input/output (I/O) operation to the Peruse command file, file-system error *number* occurred.

Effect. Peruse does not read the next command line.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

INVALID JOB NUMBER SPECIFIED

Cause. The job number entered is outside the range 1 through 4095.

Effect. Peruse does not execute the command.

Recovery. Check your job number, and then reenter your command.

JOB ALREADY IS PART OF A BATCH JOB

Cause. You attempted to link a job to a batch job to which it already belongs.

Effect. The operation is ignored.

Recovery. None is required.

JOB DOES NOT BELONG TO CURRENT USER

Cause. The job number in the JOB command does not belong to the user, or the user is not authorized to access the job.

Effect. The command is ignored.

Recovery. Check your job number, and then reenter your command.

JOB IS NOT PART OF A BATCH JOB

Cause. You attempted to unlink a job from a batch job when they were not linked.

Effect. The operation is ignored.

Recovery. This message is informational only; no corrective action is needed.

KEY ATTRIBUTES OF JOB DON'T MATCH THOSE OF BATCH JOB

Cause. There was an attempt to link to a batch job a job whose key attributes do not match those of the batch job. (These key attributes are owner, form, device, and jobid.)

Effect. The operation is ignored.

Recovery. Use the JOB command to change the key attributes of the job to match those of the batch job, or use the BATCH command to change the key attributes of the batch jobs to those of the job to be linked.

LINE TOO LONG, TRUNCATED

Cause. The command line exceeded 132 characters.

Effect. The spooler executes only the first 132 characters of your command.

Recovery. Reenter the Peruse commands that were not executed.

NO CURRENT JOB

Cause. A DELETE command was given, but no current job was defined.

Effect. Peruse ignores your command.

Recovery. Peruse requires that you specify the current job before deleting it. Use the JOB command to specify the current job, and then use the DELETE command to delete it.

NO JOBS AVAILABLE

Cause. A command that refers to the default job was entered, but no job was found in the spooler.

Effect. The spooler ignores the command.

Recovery. Respool the job, and then reenter your command.

NO OLD FIND STRING TO USE FOR COMPARES

Cause. A FIND command was entered without a *string* parameter, and no previous FIND command with a *string* parameter had been entered.

Effect. The spooler ignores the command.

Recovery. Reenter the FIND command with a *string* parameter.

NO ROOM FOR ENTRY IN SPOOLER TABLES

Cause. A LOC command was entered that specified a new spooler location, but the spooler table for locations is already full.

Effect. The spooler ignores the command.

Recovery. Warmstart the spooler with more locations configured and specify the REBUILD option to the SPOOL command. If you cannot warmstart the spooler, delete any locations that are not used.

NO SUCH JOB

Cause. A JOB command was entered with a job number that didn't match any of your jobs in the spooler.

Effect. The spooler ignores the command.

Recovery. Check your job number, and then reenter the command.

```
NOT ALLOWED ON PART OF A BATCH JOB
```

Cause. You attempted an operation on a job that is part of a batch job.

Effect. The operation is ignored.

Recovery. Either retry the operation on the batch job, or unlink the job from the batch job and then retry the operation.

```
OPEN OF SPOOL SUPERVISOR FAILED, FILE ERROR: number
```

Cause. A Guardian OPEN procedure encountered a file-system error.

Effect. Peruse cannot open the spooler supervisor process specified in the command to run Peruse or the Peruse OPEN command. (The spooler supervisor process is \$SPLS or a name given by *supervisor*.)

Recovery. Verify that you specified the correct spooler supervisor. If you did not, reexecute your command. If you did, refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for the file-system error *number*.

If the error is file-system error 14, the spooler has no supervisor. Bring the system up again.

If the error is file-system error 32 or 34, the supervisor has too many opens at one time. Wait until some operations have finished, and then retry the operation.

```
'OUT' FILE IS FULL: number
```

Cause. The OUT file specified in the Peruse command is full. *number* specifies the file-system error number.

Effect. The Peruse command cannot write in the OUT file.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

```
'OUT' FILE MUST BE EDIT FILE
```

Cause. EDIT was specified in a LIST command, and the OUT file was not an EDIT file.

Effect. The command is not executed.

Recovery. Specify an EDIT file as the OUT file, and then retry the command.

'OUT' FILE MUST BE PROCESS OR SPOOLER JOB FILE

Cause. SPOOL or COMPRESS was specified in a LIST command, and the OUT file was not a process or a spooler job file.

Effect. The command is not executed.

Recovery. Specify a process or a spooler job file as the OUT file, and then retry the command.

PROGRAM AND SPOOL LIBRARY VERSIONS DIFFER

Cause. The current version of Peruse does not match the version of the spooler library procedures.

Effect. Peruse abends.

Recovery. Ask your system operator to match the versions. A new system-generation procedure might be required.

PURGE ERROR: *number*

Cause. The file-system error *number* occurred during the execution of a PURGE command.

Effect. The operation terminates.

Recovery. For corrective action of the file-system error *number*, see the *Guardian Procedure Errors and Messages Manual*.

SPOOLER SUPERVISOR MUST BE A PROCESS

Cause. You started Peruse or invoked the OPEN command, and the specified *supervisor* was not a process name.

Effect. If Peruse was being started, it successfully starts. If the OPEN command was invoked, the command fails.

Recovery. If you need to specify *supervisor*, give the name of a spooler supervisor process, and then retry the command.

SUBCOMMAND VALUE INVALID

Cause. The *report-name* parameter of the REPORT command included one or more invalid characters.

Effect. Peruse ignores this command.

Recovery. Reenter the command, specifying a *report-name* that contains only letters, digits, and spaces.

THE JOB IS CURRENTLY BEING PRINTED

Cause. The command you entered would change the attributes of a current job, but your job is already in the print state.

Effect. The spooler ignores your command.

Recovery. To execute your command, put the job on hold. (This action deletes the job from the print queue.) Reissue your command, and then remove the hold; the job will enter the device queue again and print completely.

USERNAME NOT FOUND

Cause. The OWNER command has an incorrect *group-name.user-name*. Either the *group-name* or the *user-name* does not exist on this system, or you put a space between the *group-name* and the *user-name*.

Effect. The spooler does not execute your command.

Recovery. Reenter your command with the correct *group-name.user-name*. Place a period between *group-name* and *user-name*, and leave no spaces on either side of the period.

Peruse Messages

C Spoolcom Messages

This appendix describes the error messages that Spoolcom writes to your terminal or to its OUT file. For additional information on Spoolcom, see [Section 4, Spoolcom](#).

```
CAN'T OPEN OUT FILE:  number
```

Cause. Spoolcom could not open the OUT file.

Effect. Spoolcom reports a file-system error and uses the home terminal for the OUT file.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *number*.

```
DUPLICATE QUALIFIER FOUND
```

Cause. A JOB command was issued containing more than one occurrence of a specific parameter.

Effect. Spoolcom ignores your command.

Recovery. Reissue the JOB command with only one occurrence of that parameter.

```
EDITREAD ERROR:  number
```

Cause. A file-system error occurred when Spoolcom attempted to read an EDIT format command file.

Effect. Spoolcom abends.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *number*.

```
EDITREADINIT ERROR number
```

Cause. A file-system error occurred when an attempt was made to initialize a command file in EDIT format.

Effect. Spoolcom abends.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *number*.

ENTRY IN USE

Cause. The spooler rejected an attempt to delete a spooler component because the component was already in use. Before deleting a location, first delete all jobs associated with this location. Before deleting a device, first delete all locations using this device. Before deleting a print process, first delete all processes associated with this print process.

Effect. Spoolcom ignores the command.

Recovery. Execute another command.

ENTRY NOT FOUND

Cause. The job, device, process, location, batch job, or font referred to is unknown to the spooler.

Effect. Spoolcom ignores the command.

Recovery. Check your command for a misspelled name or an incorrect number, and then reexecute your command.

ERROR IN WRITING TO SPOOLER *number*

Cause. A file-system error occurred when Spoolcom attempted to write to the supervisor.

Effect. Spoolcom ignores the command.

Recovery. Do a TACL STATUS command and determine what happened to the spooler supervisor.

Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *number*.

ILLEGAL QUALIFIER LIST

Cause. There was a syntax error in the qualifier or qualifier list in the JOB command.

Effect. Spoolcom ignores the command.

Recovery. Reenter the command with the correct syntax.

INPUT FILE ERROR: *number*

Cause. A file-system error occurred in the Spoolcom command file.

Effect. Spoolcom abends.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *number*.

INVALID COMMAND

Cause. A command contained incorrect syntax.

Effect. Spoolcom ignores the command.

Recovery. Reexecute the command with the correct syntax.

INVALID COMMAND PARAMETER

Cause. A command was issued with a command parameter that was either incorrect or in the wrong syntax.

Effect. Spoolcom ignores the command.

Recovery. Reexecute the command with the correct command parameter.

INVALID LIST FILE

Cause. An illegal name was specified as the OUT file.

Effect. Spoolcom abends.

Recovery. Correct the syntax of the OUT file name, and then reissue the command.

INVALID OPERATION IN CURRENT STATE

Cause. A command was entered that cannot be executed while the spooler component is in its present state.

Effect. Spoolcom ignores the command.

Recovery. Refer to the tables in Section 3, Spoolcom, to determine the correct state required for the requested operation. Put the spooler component into the correct state, and then reissue the command.

INVALID PROCESS NAME

Cause. A process name was specified that has incorrect syntax.

Effect. Spoolcom ignores the command.

Recovery. Correct the syntax, and then reexecute the command.

INVALID SUBCOMMAND

Cause. A Spoolcom command has an illegal subcommand after the comma.

Effect. Spoolcom executes the command but not the subcommand portion.

Recovery. Either use the correct subcommand for this command, or enter a command that uses the subcommand you want.

INVALID SUBCOMMAND PARAMETER

Cause. A Spoolcom command was entered with a subcommand parameter that has an illegal value or incorrect syntax.

Effect. Spoolcom ignores the command.

Recovery. Check the syntax of the subcommand for characters that are not letters, digits, or spaces. Then reenter the command.

JOB ALREADY IS PART OF A BATCH JOB

Cause. You attempted to link a job to a batch job to which it already belongs.

Effect. The operation is ignored.

Recovery. This is informational only; no corrective action is needed.

JOB *number* IN USE

Cause. An attempt was made to delete a job that was either currently printing or still spooling (in the OPEN state).

Effect. Spoolcom ignores your command.

Recovery. Put the job on hold. If it was printing, you can delete it right away. If it is an open job, you can delete it after it finishes spooling.

JOB *number* INVALID OPERATION IN CURRENT STATE

Cause. A JOB command was entered that cannot be executed while your job is in its present state.

Effect. Spoolcom ignores the command.

Recovery. Refer to Section 4, Spoolcom for a list of the valid job states for the JOB command. Determine the correct state required for the requested operation. Then put the job into the correct state and reexecute the command.

JOB IS NOT PART OF A BATCH JOB

Cause. You attempted to unlink a job from a batch job, but they were not linked.

Effect. The operation is ignored.

Recovery. This message is informational only; no corrective action is needed.

JOB *number* NOT FOUND

Cause. More than one subcommand was entered for a JOB command with qualifiers, and an earlier subcommand deleted a qualified job.

Before Spoolcom executes a subcommand, it first uses the specified qualifiers to prepare a list of jobs; it does not recompile a new list of jobs for each subcommand. As a result, it displays this message for each deleted job when it cannot perform a later subcommand on it.

Effect. Spoolcom does not execute the subsequent subcommand or anything following it.

Recovery. This message is informational only; no corrective action is needed.

JOB *number* SECURITY VIOLATION

Cause. A Spoolcom JOB command was entered that would affect jobs other than your own. Such commands are restricted to authorized users such as system operators (user ID 255,*n*) and group managers (user ID *n*, 255).

Effect. Spoolcom ignores the command.

Recovery. Informational message only; no corrective action is needed.

KEY ATTRIBUTES OF JOB DON'T MATCH THOSE OF BATCH JOB

Cause. You tried to link a job to a batch job, but the job attributes (owner, form, device, and/or jobid) do not match those of the batch job.

Effect. The operation is ignored.

Recovery. Change the key attributes of the job to match those of the batch job.

LINE TOO LONG, TRUNCATED

Cause. A Spoolcom command line was longer than 132 characters.

Effect. Spoolcom attempts to execute the first 132 characters of the command line then ignores any characters beyond those.

Recovery. Enter a Spoolcom command to complete the unexecuted portion of this command.

MISSING SUBCOMMAND PARAMETERS

Cause. A subcommand keyword was entered but did not specify any value, and there was no default value.

Effect. Spoolcom ignores the command.

Recovery. Supply the missing value and retry the command.

NOT ALLOWED ON PART OF A BATCH JOB

Cause. You attempted an operation on a job that was part of a batch job.

Effect. The operation is ignored.

Recovery. Perform the operation on the batch job, or unlink the job from the batch job and retry the operation.

NULL DATE RANGE SPECIFIED

Cause. A null date range was specified in a JOB qualifier list.

Effect. Spoolcom executes the command up to this point and then stops.

Recovery. Enter a Spoolcom command that completes the unexecuted portion of this command.

SECURITY COMMAND

Cause. You entered a Spoolcom command that is restricted to authorized users such as system operators (user ID 255,*n*) and group managers (user ID *n*,255).

Effect. Spoolcom ignores the command.

Recovery. Informational message only; no corrective action is needed.

Spoolcom ATTACHED TO UNKNOWN TYPE PROCESS

Cause. The last OPEN command specified a process other than a spool supervisor.

Effect. Spoolcom ignores the command.

Recovery. Correct the supervisor name.

SPOOLCOM DOES NOT SUPPORT > 65500 PAGES USE PERUSE

Cause. The user attempted to access pages beyond the 65500 page number limit for a job.

Effect. The Spoolcom command fails.

Recovery. Use Peruse, which allows access to pages in the range 0 through 999,999.

SPOOLER FILE ERROR *number*

Cause. A file-system error occurred on the file to the supervisor; the supervisor might not have opened the file.

Effect. Spoolcom closes the file to the supervisor.

Recovery. If you are sure the file was opened, check the status of the supervisor or open another supervisor. Refer to the *Guardian Procedure Errors and Messages Manual* for file-system error *number*.

SPOOLER NAME MISSING

Cause. An OPEN command did not specify the required supervisor name.

Effect. Spoolcom ignores the command.

Recovery. Specify the supervisor name.

SPOOLER NOT OPEN

Cause. A command was issued before the spooler was opened.

Effect. Spoolcom ignores the command.

Recovery. Open the spooler before issuing another Spoolcom command.

SPOOLER OPEN ERROR *number*

Cause. A Guardian file-system error occurred when the application attempted to open a file to the supervisor.

Effect. The spooler has no supervisor at this time. Spoolcom ignores the command to open a file, and the file to a previous supervisor (if any) is closed.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *number*.

Remove the error condition before you try to open a file to the supervisor again.

If file-system error 14 occurred, you must bring up the supervisor before the operation can succeed.

If file-system error 32 or 34 occurred, the supervisor has too many opens at one time. Wait until some operations have finished and then retry the operation.

TABLE LIMIT EXCEEDED

Cause. A command to add an entry was issued, but the maximum number of devices, locations, or processes specified when the spooler was coldstarted was already reached.

Effect. Spoolcom ignores the command.

Recovery. Drain the spooler, and then add more of the needed components before warm starting.

THERE ARE NO JOBS CURRENTLY OPEN

Cause. A command was issued to list the open jobs for a collector that has no jobs currently open.

Effect. The operation is ignored.

Recovery. Informational message only; no corrective action is needed.

WARNING: USERID NOT DEFINED

Cause. Either the user ID was not recognized on the system or network or \$SYSTEM.SYSTEM.USERID cannot be accessed.

Effect. Spoolcom executes the command.

Recovery. Specify a valid user ID.

D Font Utility Messages

This appendix describes the error messages that the Font utility displays to the OUT file of the command used to run Font (the default is the home terminal).

```
^ ** ERROR Decimal Number too big or invalid
```

Cause. The number specified after the # character is invalid. It is greater than 255, less than 0, or does not consist of the digits 0 through 9. It must be a decimal number in the range 0 through 255.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the byte value specified after the # character. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following Spoolcom command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Device Name invalid
```

Cause. The device name specified is not a valid device name. The device name must be a valid Guardian file name, process name, or device name in the form:

```
[\system.] [$name1.] [#name2.] [name3]
```

One of these, *name1*, *name2*, or *name3*, must be specified.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the device name. For syntax descriptions of process or device names, see the *Guardian Procedure Calls Reference Manual*.

In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR FONT name greater than 16 chars
```

Cause. The *font-name* parameter specified in the FONT directive or after a location in the locations section was more than 16 characters long.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Shorten the font name so that it does not exceed 16 characters. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR FONT name must be contiguous
```

Cause. Either the *font-name* parameter specified in the FONT directive contains spaces, or a comment after *font-name* did not begin with an exclamation point (!).

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Either remove blanks from the font name or put the ! character before any comments on this line. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR %nnnnnn from PRINTREAD
```

Cause. The Font utility received error *nnnnnn* from a PRINTREAD call when trying to read the next record from the spooler job file (file code 129) specified as a source file in a SOURCE directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLCONTROL
```

Cause. An internal Font utility error occurred. Error *nnnnnn* was returned from a SPOOLCONTROL procedure call. The Font utility was attempting to place a CONTROL in the font job that it was currently spooling.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLCONTROLBUF
```

Cause. An internal Font utility error occurred. Error *nnnnnn* was returned from a SPOOLCONTROLBUF procedure call. The Font utility was attempting to place a CONTROLBUF in the font job that it was currently spooling.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLEND
```

Cause. An internal FONT utility error occurred. Error *nnnnnn* was returned from a SPOOLEND procedure call. The Font utility was attempting to close the font job that it was currently spooling.

Effect. The FONT utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLJOBNUM
```

Cause. An internal Font utility error occurred. Error *nnnnnn* was returned from a SPOOLJOBNUM procedure call. The Font utility was attempting to get a job number assigned to the font job that it was currently spooling.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLSETMODE
```

Cause. An internal Font utility error occurred. Error *nnnnnn* was returned from a SPOOLSETMODE procedure call. The Font utility was attempting to place a SETMODE in the font job that it was currently spooling.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLSTART
```

Cause. An internal Font utility error occurred. Error *nnnnnn* was returned from a SPOOLSTART procedure call. The Font utility was attempting to start a level-3 spooling connection with the collector to spool a font job.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
** ERROR : %nnnnnn from SPOOLWRITE
```

Cause. An internal Font utility error occurred. Error *nnnnnn* was returned from a SPOOLWRITE procedure call. The Font utility was attempting to send font job data to the collector that was receiving a new font job.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Contact your service representative. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Hex Number too big or invalid
```

Cause. The number specified after the H character is invalid. It is greater than HFF, or less than 0, or does not consist of the hexadecimal digits 0 through F. It must be a hexadecimal number in the range H0 through HFF.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the byte value specified after the H character. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.


```
^ ** ERROR Invalid beginning-count
```

Cause. The value of *beginning-count* specified in the SOURCE directive is invalid. *beginning-count* must be a number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the value for *beginning-count* in the SOURCE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid CONTROL Operation
```

Cause. The *operation* value specified in the CONTROL directive is invalid. *operation* must be a number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the value for *operation* the CONTROL directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid CONTROL parameter
```

Cause. The *param* value specified in the CONTROL directive is invalid. *param* must be a positive decimal number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the *param* value in the CONTROL directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid CONTROLBUF Operation
```

Cause. The *operation* value specified in the CONTROLBUF directive is invalid. *operation* must be a number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the *operation* value in the CONTROLBUF directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid directive
```

Cause. A directive other than a Font or END-FONT directive followed a FONT directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct any misspelled directives or add a FONT or END-FONT directive after the FONT directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR Invalid Location destination
```

Cause. The destination name of a specified location is invalid. The correct syntax for a location name is *#group.destination*. The value of *destination* must be from 1 through 8 alphanumeric characters, with the first character being a letter.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the destination name specified in the location. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR Invalid Location group
```

Cause. The group name of a specified location is invalid. The correct syntax for a location name is *#group.destination*. The value of *group* must be from 1 through 7 alphanumeric characters, with the first character being a letter.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the group name specified in the location. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR Invalid SETMODE Operation
```

Cause. The *function* value specified in the SETMODE directive is invalid. The value must be a number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the *function* value in the SETMODE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid SETMODE parameter
```

Cause. The *param1* or *param2* value specified in the SETMODE directive is invalid. *param1* or *param2* must be a number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the values for *param1* and *param2* in the SETMODE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid SOURCE Filename
```

Cause. The *source-file* filename specified in the SOURCE directive is an invalid file name.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct *source-file* in the SOURCE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Invalid trailing-count
```

Cause. The *trailing-count* specified in the SOURCE directive is invalid. *trailing-count* must be a number in the range 0 through 32767.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the value for *trailing-count* in the SOURCE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Missing CONTROL Operation
```

Cause. The *operation* value was not found in a CONTROL directive in a font description.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify an *operation* value in the CONTROL directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Missing CONTROL Parameter
```

Cause. The *param* value was not found in a CONTROL directive in a font description.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify a *param* value in the CONTROL directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Missing CONTROLBUF Operation
```

Cause. The *operation* value was not found in a CONTROLBUF directive in a font description.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify an *operation* value in the CONTROLBUF directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Missing FONT Name
```

Cause. The *font-name* parameter was not found in a FONT directive or the *font-name* was not found after a location name in the locations section.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify a valid font name. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
^ ** ERROR Missing Quotes
```

Cause. A data line was specified for the CONTROLBUF directive that did not begin with any of these characters: ", #, %, H, !, or ?.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Use quotation marks (") to delimit actual text data or use one of the other symbols as the first nonblank character on the line. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be

deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Missing SETMODE Operation
```

Cause. The *function* value was not found in a SETMODE directive in a font description.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify a *function* value in the SETMODE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Missing SETMODE Parameter
```

Cause. The *param1* or *param2* value was not found in a SETMODE directive in a font description.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify both *param1* and *param2* in the SETMODE directive. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.


```
^ ** ERROR Missing SOURCE Filename
```

Cause. The *source-file* filename was not found in a SOURCE directive in a font description.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Specify a filename for *source-file*. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR Octal Number too big or invalid
```

Cause. The number specified after the % character is invalid. It is greater than %377, or less than 0, or does not consist of the digits 0 through 7. It must be a decimal number in the range %0 through %377.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Correct the byte value specified after the % character. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the FONT utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR nnnn on file-name
```

Cause. The Font utility received the file-system error *nnnn* when trying to access the source file (*file-name*) specified in the SOURCE directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *nnnn*. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR ( OPEN ) nnnn on file-name
```

Cause. The Font utility could not open the source file (*file-name*) specified in the SOURCE directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *nnnn*. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR ( OPEN^FILE ) nnnn on file-name
```

Cause. The Font utility could not open the source file (*file-name*) specified in the SOURCE directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *nnnn*. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
^ ** ERROR ( READ^FILE ) nnnn on file-name
```

Cause. The Font utility could not read the source-file (*file-name*) specified in the SOURCE directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Refer to the *Guardian Procedure Errors and Messages Manual* for corrective action for file-system error *nnnn*. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

There might be a job that was created in the spooler that has its ABNORMAL flag set to YES because the Font utility did not close the job properly. This job should be deleted. You can find jobs that the Font utility created with the following SPOOLCOM command:

```
SPOOLCOM ; JOB (REPORT LASER FONT JOB)
```

If the job has an X in the FLAGS column and the TIME field corresponds to the time that you ran the Font utility, delete this job.

```
*** ERROR Unable to find ?START-FONT directive in file-name
```

Cause. The Font utility did not find a START-FONT directive anywhere in the script file. The Font utility ignores all text up to the START-FONT directive.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Put a START-FONT directive as the first directive in your script file. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
FILE OPEN ERROR - file - buffer too small (518)
```

Cause. You specified an OUT *listfile* that was an EDIT file. The FONT utility cannot direct output to an EDIT file.

Effect. The Font utility aborts.

Recovery. Specify another type of file as the list file.

```
FILE OPEN ERROR - file - nonexistent device (14)
```

Cause. The process you specified as the supervisor process does not exist.

Effect. The Font utility aborts.

Recovery. Specify the name of an existing supervisor process.

```
FILE OPEN ERROR - file - nonexistent file (11)
```

Cause. You specified either a script file or a list file that does not exist.

Effect. The Font utility aborts.

Recovery. Specify an existing file for the script file or the list file.

```
**SPOOLERCOMMAND Error :%14010 ( Cannot Add this to Tables )
**Cmd: ccc ( "command-name" ) Subcmd: sss ("subcommand-name"
)
```

Cause. The Font utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. The value of *sss* is the subcommand number issued. *subcommand-name* is a text description of the subcommand.

The Font utility tried to add a job, location, font, or print process that exceeded the spooler limits for this item.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Either delete unnecessary spooler items to free up space or increase the spooler limits. See *coldstart-parameters* in Appendix A, SPOOL Program. Remove any unwanted font jobs and configuration changes. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
**SPOOLERCOMMAND Error :%14011 ( Cannot find this Entry )
**Cmd: ccc ( "command-name" ) Subcmd: sss ("subcommand-name"
)
```

Cause. The Font utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. The value of *sss* is the subcommand number issued. *subcommand-name* is a text description of the subcommand.

A font name that does not exist was specified in the locations section. The script-file line number is displayed before this error message.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Make sure that the font name specified in this section matches a font name specified in a Font directive or matches an existing spooler font name. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
**SPOOLERCOMMAND Error :%14013 ( Presently in use )
**Cmd: ccc ( "command-name" ) Subcmd: sss ("subcommand-name"
)
```

Cause. The Font utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. The value of *sss* is the subcommand number issued. *subcommand-name* is a text description of the subcommand.

The Font utility tried to delete a font job that was in the open or print state because a new job was to be assigned to an existing font name.

Effect. The Font utility aborts. Any font jobs created and configuration changes made to this point remain.

Recovery. Change the state of the font job currently assigned to the font to hold or ready. Remove any unwanted font jobs and configuration changes. In most cases, you can fix the script file and rerun the Font utility without undoing any configuration changes that the Font utility made in the spooler.

```
**SPOOLERCOMMAND Error :%14014 ( Security Violation )
**Cmd: ccc ( "command-name" ) Subcmd: sss ("subcommand-name"
)
```

Cause. The Font utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. The value of *sss* is the subcommand number issued. *subcommand-name* is a text description of the subcommand.

You tried to use the Font utility and you are neither the owner of the spooler nor a system operator (user ID 255,*n*).

Effect. The Font utility aborts.

Recovery. Log on as either the spooler owner or a system operator (user ID 255,*n*) to use the Font utility.

```
**SPOOLERSTATUS Error :%003002 ( error-description )
**Cmd: ( "command-name" , "subcommand-name") Scan: sss
```

Cause. The Font utility received an error from the SPOOLERSTATUS procedure. The value of *error-description* is a text description of the error. The value of *command-name* is a text description of the command. The value of *subcommand-name* is a text description of the subcommand. The value of *sss* is the SCAN^TYPE value passed to SPOOLERSTATUS.

The specified supervisor process is an existing process, but it is not a spooler supervisor process.

Effect. The Font utility aborts.

Recovery. Specify an existing supervisor process.

```
**SPOOLERSTATUS Error :%14006 ( End of Spoolerstatus hits )  
**Cmd: ( "command-name" , "subcommand-name") Scan: sss
```

Cause. The Font utility received an error from the SPOOLERSTATUS procedure. *command-name* is a text description of the command. *subcommand-name* is a text description of the subcommand. *sss* is the SCAN^TYPE value passed to SPOOLERSTATUS.

Either the spooler does not have a collector process configured or none of the collector processes is in the active state.

Effect. The Font utility aborts.

Recovery. Either configure a collector in the spooler or change the collector status to active to allow the Font utility to spool a font job.

```
** WARNING ** Can't find Device: device to add to  
Location: location
```

Cause. The locations section (following a START-LOCATIONS directive) specified a device name that does not exist.

Effect. The Font utility uses the first device it finds in the spooler. If there are no devices defined in the spooler, the location that is created is not linked to a device.

Recovery. Specify an existing spooler device name.

E RPSetup Utility Messages

The appendix describes the error messages that the RPSetup utility displays to the home terminal of the command used to run RPSetup.

```
FILE OPEN ERROR - supervisor - nonexistent device (14)
```

Cause. The specified remote or local spooler does not exist.

Effect. The FONT utility aborts.

Recovery. Specify the name of an existing supervisor process.

```
**SPOOLERCOMMAND Error :%14010 (Cannot Add this to Tables)
**Cmd: ccc ( "command-name" ) Subcmd: sss ( "subcommand-name"
)
```

Cause. The RPSetup utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. *sss* is the subcommand number issued. The value of *subcommand-name* is a text description of the subcommand.

The RPSetup utility tried to add a device, location, or print process that exceeded the spooler limits for this item.

Effect. The RPSetup utility aborts. Any configuration changes made up to this point remain.

Recovery. Either delete unnecessary spooler items to free up space or increase the spooler limits. See Appendix A for *cold-start-parameters*. In most cases, you can rerun the RPSetup utility without undoing any configuration changes that RPSetup previously made in the spooler.

```
**SPOOLERCOMMAND Error :%14012 (Wrong State for this Command)
**Cmd: ccc ( "command-name" ) Subcmd: sss ("subcommand-name"
)
```

Cause. The RPSetup utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. The value of *sss* is the subcommand number issued. The value of *subcommand-name* is a text description of the subcommand.

RPSetup tried to update a device configuration of a device in the waiting state. This error can occur when you specify a remote spooler and a local spooler on the same system.

Effect. The RPSetup utility aborts. Any configuration changes made up to this point remain.

Recovery. Change the device state of the indicated device to offline by issuing the following command:

```
SPOOLCOM; DEV device-name, DRAIN
```

Then rerun RPSetup. In most cases, you can rerun the RPSetup utility without undoing any configuration changes that RPSetup previously made in a spooler.

```
**SPOOLERCOMMAND Error :%14013 (Presently in use)
**Cmd: ccc ( "command-name" ) Subcmd: sss ( "subcommand-name"
)
```

Cause. The RPSetup utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. The value of *command-name* is a text description of the command. The value of *sss* is the subcommand number issued. The value of *subcommand-name* is a text description of the subcommand.

The RPSetup utility tried to delete a device or location that is currently in use.

Effect. The RPSetup utility aborts. Any font jobs created or configuration changes made up to this point remain.

Recovery. Put all jobs in the remote spooler and local spooler linked to devices that are collector processes on the remote spooler. Ensure that no jobs are actively being transferred from the local spooler to the remote spooler when running RPSetup to update the configuration of links to remote devices.

```
**SPOOLERCOMMAND Error :%14014 (Security Violation)
**Cmd: ccc ( "command-name" ) Subcmd: sss ( "subcommand-name"
)
```

Cause. The RPSetup utility received an error from the SPOOLERCOMMAND procedure. The value of *ccc* is the command number issued. *command-name* is a text description of the command. *sss* is the subcommand number issued. *subcommand-name* is a text description of the subcommand.

You tried to use the RPSetup utility and you are neither the owner of the spooler nor a system operator (user ID 255,*n*).

Effect. The RPSetup utility aborts.

Recovery. Log on as either the spooler owner or a system operator (user ID 255,*n*) user to use the RPSetup utility.

```
**SPOOLERSTATUS Error :%003002 ( error-description )
**Cmd: ( "command-name" , "subcommand-name" ) Scan: sss
```

Cause. The RPSetup utility received an error from the SPOOLERSTATUS procedure. *error-description* is a text description of the error. *command-name* is a text description of the command. *subcommand-name* is a text description of the subcommand. *sss* is the SCAN^TYPE value passed to SPOOLERSTATUS.

The specified remote or local spooler is an existing process, but it is not a spooler supervisor process.

Effect. The RPSetup utility aborts.

Recovery. Specify the name of an existing supervisor process.

```
**SPOOLERSTATUS Error :%14006 (End of Spoolerstatus hits)
**Cmd: ( "command-name" , "subcommand-name" ) Scan: sss
```

Cause. The RPSetup utility received an error from the SPOOLERSTATUS procedure. The value of *command-name* is a text description of the command. The value of *subcommand-name* is a text description of the subcommand. The value of *sss* is the SCAN^TYPE value passed to SPOOLERSTATUS.

Either the remote spooler does not have a collector process configured or none of the collector processes is in the active state.

Effect. The RPSetup utility aborts.

Recovery. Either configure a collector in the remote spooler or change the collector status to active to allow the FONT utility to spool a font job.

F Using FASTP

FASTP is the print process released with the spooler. Print processes retrieve the output stored on disk by a collector and print it on a device. Each print device in the spooler subsystem has a print process associated with it.

Topics described in this appendix include the following:

- [Configuring FASTP](#) on page F-1
- [Tracing FASTP](#) on page F-2
- [Extended Configuration With FASTCNFG](#) on page F-5

Configuring FASTP

FASTP is configured as a print process to the spooler subsystem using the Spoolcom PRINT command, which is summarized below.

```
PRINT process-name           ! suggest $SPLA or SPLX
, BACKUP backup-cpu         ! -1:15
, CPU cpu                   ! 0:15
, DEBUG [ OFF ]
, DELETE
, FILE program-filename     ! Should be FASTP
, PARM parameter           ! 0 (unused by FASTP)
, PRI execution-priority    ! 1:199
, START
, STATUS [ / OUT filename / ] [ DETAIL ]
, XREF [ / OUT filename / ]
```

FASTP does not run as a process pair; therefore, you should configure a backup processor so the supervisor can run the print process in an alternate processor.

The subcommand PARM is interpreted by FASTP as described under the DEV command in Section 4, Spoolcom.

FILE *program-filename* should be the FASTP object file supplied with the spooler subsystem.

FASTP can handle up to 32 concurrent jobs on as many active devices. In this context, an active device is one that FASTP has been instructed to open by the spooler supervisor. FASTP opens each device with write-only, exclusive access.

If the spooler supervisor tries to instruct FASTP to open a thirty-third device, FASTP will reject the open request with spooler error code %13004. This will cause the device to enter the DEV ERROR 4 state. The supervisor will periodically retry the open request.

The FASTP print process supports CONTROLBUF 101, 103, and 113 calls in font jobs as described with the CONTROLBUF directive described in Section 5, Font Utility.

Note. A CONTROLBUF 113 must be part of a font job to be processed by the FASTP print process.

Tracing FASTP

Tracing is performed by a FASTP print process for a device. The FASTP print process writes trace data to a trace file. Trace data includes information sent between the FASTP print process and the device along with information about other processing events that affect the device.

- Tracing for a device can be enabled or disabled at any time.
- A different trace file or set of trace files can be specified for each device.

To obtain trace information for a device:

1. Create the trace file (or files). Trace files must be created before tracing is enabled.
2. Specify the trace filename (or filenames) in the device entry in the FASTCNFG file. Trace filenames must be valid Guardian filenames.
3. Enable tracing for the device and force the FASTP print process to read the FASTCNFG file to determine the trace filename.

Creating Trace Files

Trace files have the following characteristics:

- A maximum of eight trace files can be specified for each device.
- If only one trace file is specified, tracing is stopped when the file is full.
- If more than one file is specified, tracing wraps around the sequence of files. When a file becomes full, the print process purges the data from the next file and begins writing to it; when the last file in the sequence is full, the print process goes to the first file in the sequence, and so on.

Use the File Utility Program (FUP) to create each trace file as follows:

```
FUP CREATE $volume.subvolume.file-ID,TYPE E,REC 4000,EXT
(n,n)
```

\$volume.subvolume.file-ID

is the name of the trace file.

TYPE E

specifies an entry-sequenced Enscribe file.

```
REC 4000
```

specifies a record size of 4000 bytes.

```
EXT (n,n)
```

specifies the primary and secondary file extent sizes and depends on the amount of file space required. An extent consists of 2048 bytes.

Specifying Trace Files for a Device

One or more trace files for a device are specified with the SHADOW attribute, which is included as part of the device entry in the FASTCNFG file:

```
DEVICE $device-name
CONTYPE = connection-type
SHADOW = $volume.subvolume.file-ID
```

Because there can be a maximum of eight trace files for each device, there are eight SHADOW attribute keywords:

```
DEVICE $device-name
CONTYPE = connection-type
SHADOW1 = $volume.subvolume.file-ID1
SHADOW2 = $volume.subvolume.file-ID2
SHADOW3 = $volume.subvolume.file-ID3
SHADOW4 = $volume.subvolume.file-ID4
SHADOW5 = $volume.subvolume.file-ID5
SHADOW6 = $volume.subvolume.file-ID6
SHADOW7 = $volume.subvolume.file-ID7
SHADOW8 = $volume.subvolume.file-ID8
```

Enabling Tracing

Bit 1 of the SPOOLCOM PARM (a 16-bit binary string) device attribute enables and disables tracing by a product-specific version of the FASTP print process.

Setting bit 5 of the 16-bit binary string to 1 forces a FASTP print process to read a FASTCNFG configuration file where, in addition to other required information about a device, it can find the name of the trace file that it is supposed to use for the device.

- To enable tracing: Set bit 1 of the 16-bit binary string to 1.
- To disable tracing: Set bit 1 of the 16-bit binary string to 0.

If any other bits must be set, they must all be set with one PARM entry. Do not use multiple PARM entries for one device.

If only bits 1 and 5 are set to 1, PARM would have a decimal value of 17408:

```
DEV $device-name PARM 17408
```

Note. Trace files must be created before tracing is enabled.

If tracing is enabled for a device and either no SHADOW file has been specified or a problem is encountered when writing to a SHADOW file, tracing for that device is disabled.

Reading Trace Output

Each trace event causes two or more records to be written to a trace file.

The first record (the identifier record) contains

- The type of event
- The time event occurred
- An abbreviated stack trace for the procedure that produced the event

The second record (the data record) contains data related to the event. If the amount of data exceeds the trace record size, the data is written as two or more records.

Because the trace file is an entry-sequenced Enscribe file, it is recommended that you use a FUP COPY command to list the trace information, for example:

```
TACL 1>FUP COPY trace-file-name,$S.#location,H
File Utility Program - T6553D45 - (02FEB2001)      System \YES
Copyright Tandem Computers Incorporated 1981, 1983, 1985-1996

$DATA4.BGXEROX.SHADA RECORD 0 KEY 0 (%0) LEN 30 7/13/97 9:22
000: 3036 204A . . . 3920 2D20 06 Jul 97 12:25:05:39 -
00C 4F75 7470 . . . 3736 2C30 Output - 171623,130476,0
018 3037 3336 . . . 07364,156635

$DATA4.BGXEROX.SHADA RECORD 1 KEY 1 (%1) LEN 74
000: 004A 027A . . . AA33 11E0 .J.zC.....:....3
00C: 0000 0000 . . . AA33 11E0 .....H.*.....d...3
018: 0005 0000 . . . 0000 0000 .....d.....(....^
024: 0002 ..
```

The first record identifies:

- The type of event: Output
- The time the event occurred: 06 Jul 97 12:25:05:39

The second record contains actual trace data.

[Table F-1](#) summarizes types of events and data that can appear in a trace file.

Table F-1. Types of Trace File Events and Data (page 1 of 2)

Event Identifier From First Record for Trace Event	Contents of Second Record for Trace Event
Controlbuf Out	CONTROLBUF data sent to the device.
Control Out	CONTROL operation and parameter sent to the device.
Input	Data that was received from the device.
Output	Data that was written to the device.
Setmode In	SETMODE operation and two parameters received from the device.

Table F-1. Types of Trace File Events and Data (page 2 of 2)

Event Identifier From First Record for Trace Event	Contents of Second Record for Trace Event
Setmode Out	SETMODE operation and two parameters sent to the device.
Supervisor - Batch Status In	Response received from supervisor process containing status for a batch.
Supervisor - Batch Status Out	Request sent to supervisor process to solicit status of a batch.
Supervisor - Collector Status In	Response received from supervisor process containing status for a collector process.
Supervisor - Collector Status Out	Request sent to supervisor process to solicit status of a collector process.
Supervisor - Command In	Command received (buffer contents) from the supervisor process. Supervisor commands are logged only if the device is already identified. For example, an OPEN command is not usually logged.
Supervisor - DEV Status In	Response received from supervisor process containing status for a device.
Supervisor - DEV Status Out	Request sent to supervisor process to solicit status of a device.
Supervisor - Job Status In	Response received from supervisor process containing status for a job.
Supervisor - Job Status Out	Request sent to supervisor process to solicit status of a job.

Extended Configuration With FASTCNFG

The FASTP print process can use a configuration file named `FASTCNFG`.

- Instances of FASTP can share the same `FASTPCNFG` file.
- Each FASTP print process must be instructed to read the `FASTPCNFG` file when it is started.

To instruct FASTP to read the `FASTCNFG` file, set bit 5 of the `PARAM` device attribute (a 16-bit binary string) to 1.

If any other bits are set to 1, they must all be set with one `PARAM` entry. Do not use multiple `PARAM` entries for one device.

If only bit 5 is set to 1, `PARAM` would have a decimal value of 1024:

```
DEV $device-name, PARAM 1024
```

Bit 0 is the most significant bit of the word.

Creation

The default location for the file is `$$SYSTEM.SYSTEM.FASTCNFG`. If a product-specific version of the FASTP print process is looking for a FASTCNFG file, it expects the file name to be `$$SYSTEM.SYSTEM.FASTCNFG` unless the file name is explicitly defined with the following DEFINE statements:

```
SET DEFMODE ON  
ADD DEFINE = FASTCNFG, FILE $volume.subvolume.file-ID
```

A spooler subsystem uses the information in a DEFINE statement only after the subsystem is started. Therefore, if a spooler subsystem was started before the DEFINE statement was executed, you must stop and restart the entire subsystem.

Redirection

A FASTCNFG file can redirect a product-specific version of the FASTP print process from one FASTCNFG file to another. The primary FASTCNFG file is `$$SYSTEM.SYSTEM.FASTCNFG` (or the file specified in TACL DEFINE commands). A print process can be redirected to one or two subordinate files.

Security

A FASTCNFG file usually contains information that should not be accessible to all users. A file can be secured with one or both of the following mechanisms:

- Guardian file-access attributes
- Safeguard protection

For more information about file security, refer to the following manuals:

- *Guardian Procedure Calls Reference Manual*
- *Safeguard Administrator's Manual*

The file must be secured for read access by the FASTP print process.

Alteration

Device information can be added to, deleted from, or changed in a FASTCNFG file at any time. However, updated information for a device becomes effective only after a product-specific version of the FASTP print process controlling the device reads the information.

The first time FASTP opens a device, it reads the primary FASTCNFG file and saves the file timestamp and the information in the configuration entry for the device. Each time the print process opens that device, it checks the timestamp for the primary FASTCNFG file again. If the file has been updated, the print process reads the information in the configuration entry for the device and updates its own timestamp and configuration information for this device.

One of the SPOOLCOM device attributes is EXCLUSIVE [ON|OFF]. The default value is EXCLUSIVE OFF.

If EXCLUSIVE OFF is specified, a device is opened only if a job is queued for the device and FASTP does not already have the device open. The print process opens the device and sends the job to it.

If a job is queued for the device and the print process already has the device open, the print process sends the job to the device without first opening it.

If no jobs are queued for the device and the print process has the device open, it closes the device.

If EXCLUSIVE ON is specified, the print process opens the device once and keeps it open (that is, the print process does not open and close the device).

To ensure that an altered FASTCNFG file is read:

- If only a subordinate FASTCNFG file has been altered, update the timestamp of the primary FASTCNFG file as well. To change the timestamp, modify something in the file (for example, add and delete a character).
- If EXCLUSIVE ON is specified for a device, change the attribute to EXCLUSIVE OFF !. EXCLUSIVE OFF ! specifies that the print process should close the device between jobs. Therefore, before the print process opens the device for the next job, it reads the primary FASTCNFG file and checks the timestamp. If the file has been updated, the print process reads the information in the configuration entry for the device and updates its own timestamp and configuration information for the device. Once EXCLUSIVE OFF ! has been specified and a job subsequently sent to the device, EXCLUSIVE ON must be specified again.

- Use the following SPOOLCOM commands:

```
DEV $device-name,DRAIN
DEV $device-name,EXCLUSIVE OFF!
DEV $device-name,START
```

```
/* Send job to device */
```

```
DEV $device-name,DRAIN
DEV $device-name,EXCLUSIVE ON
DEV $device-name,START
```

- If EXCLUSIVE OFF ! is specified for a device, if there are many jobs in its print queue, and if it is necessary to ensure that an altered FASTCNFG file is read before all of the jobs in its print queue are sent to the device, force the print process to close and open the device by draining and starting the device. Use the following SPOOLCOM commands:

```
DEV $device-name,DRAIN
DEV $device-name,START
```

Syntax

The syntax for a FASTCNFG file is informally structured and consists of a keyword and keyword value followed by one or more pairs of attributes and attribute values:

```
keyword keyword-value
attribute = attribute-value
[ attribute = attribute-value ]...
```

Miscellaneous Syntax Elements

Miscellaneous syntax elements include:

- Case-sensitivity
- Comments
- Line continuation
- Separator characters
- Starting position relative to the beginning of a line

Case-Sensitivity

FASTCNFG file syntax is not case sensitive. However, some attribute values can be case-sensitive (for example, a password value).

Comments

Comments can be inserted anywhere in the FASTCNFG file and must follow an exclamation mark (!). Anything following an exclamation mark is considered to be a comment until another exclamation mark or the end of the line is encountered.

Line Continuation

Line continuation does not require any special characters or format as long as you do not split keywords, keyword values, attributes, or attribute values. For example:

```
!This is legal
SPOOLER
$SPLG
TIMEOUT
=
999

!This is not legal
SPOOL
ER $SPLG
TIME
OUT=
999
```

Separator Characters

Keywords and keyword values must be separated with one or more blank spaces. Attributes and attribute values must be separated with an equal (=) sign. One or more

blank spaces can be inserted before or after the = sign. Keyword and keyword value pairs and attribute and attribute value pairs must be separated with one or more blank spaces. For example:

```
!This is legal
DELAY=999

!This is legal
DELAY      =      999
```

Starting Position Relative to the Beginning of a Line

Keywords and attributes can begin in any position on a line. For example:

```
!This is legal
      DELAY=999
```

Keywords

There are three keywords:

- SPOOLER
- PROCESS
- DEVICE

`SPOOLER [\node.]$supervisor-process-name`

specifies a spooler subsystem. Everything that follows a SPOOLER keyword pertains to the spooler subsystem specified with the keyword until another SPOOLER keyword is encountered. If the node name is not supplied, the default node name for the process executing the command is substituted. The default spooler subsystem supervisor name is `$SPLS`.

`PROCESS [\node.]$process-name`

specifies the name of an existing product-specific version of a FASTP print process. Everything that follows a PROCESS keyword pertains to the print process specified with the keyword until another PROCESS keyword is encountered. If the node name is not supplied, the default node name for the process executing the command is substituted.

`DEVICE [\node.]$device-name [.#qualifier]`

specifies a device name specified to a spooler subsystem to represent a local area network (LAN) print queue. If the node name is not supplied, the default node name for the process executing the command is substituted.

Attributes

All attributes, except for the FASTCNFG attribute, are device attributes. Some device attributes (and their attribute values) can be used with any product-specific version of a FASTP print process. Other device attributes (and their attribute values) are used only

with one product-specific version of a FASTP print process. Device attributes are described in the product-specific sections of this manual.

The next example shows a device keyword and keyword value (DEVICE \$PRT) followed by two pairs of attributes and attribute values (CONTYPE = NONE).

```
DEVICE $PRT
CONTYPE = NONE
```

The LAN print queue has already been configured with SPOOLCOM and has been assigned the name \$PRT. The keyword value and attribute values tell the print process that the device being described is the device represented by the name \$PRT.

The FASTCNFG attribute is used only for FASTCNFG file redirection.

FASTCNFG File Redirection

The FASTCNFG attribute is used to redirect a print process from one FASTCNFG file to another. The attribute specifies the name of the other FASTCNFG file.

- The primary FASTCNFG file, \$SYSTEM.SYSTEM.FASTCNFG (or the file specified with the TACL DEFINE commands), is referred to as the level-1 file.
- A print process can be redirected from the primary FASTCNFG file to another FASTCNFG file. The subordinate FASTCNFG file is referred to as a level-2 file.
- A print process can be redirected from a level-2 FASTCNFG file to another FASTCNFG file. The subordinate FASTCNFG file is referred to as a level-3 file.

No more than three levels are recommended.

Redirection to a subordinate file is specified with the FASTCNFG attribute. Once a product-specific version of the FASTP print process is redirected to a subordinate FASTCNFG file, it does not return to the original file.

The syntax for the FASTCNFG file redirection attribute is as follows:

```
FASTCNFG = [\node.] [$volume.] [subvolume.] file-ID
```

```
[\node.] [$volume.] [subvolume.] file-ID
```

specifies the Guardian file name for a FASTCNFG file. If the file name is not fully qualified, the default node name, volume, and subvolume for the print process reading the file are substituted.

The FASTCNFG attribute is used with the SPOOLER and PROCESS keywords. It should not be used with the DEVICE keyword.

The syntax for specifying a subordinate FASTCNFG file in a spooler subsystem is as follows:

```
SPOOLER [\node.] $supervisor-name
FASTCNFG = [\node.] [$volume.] [subvolume.] file-ID
```

The syntax for specifying a subordinate FASTCNFG file for one print process is as follows:

```
PROCESS [\node.]$process-name
FASTCNFG = [\node.][$volume.][subvolume.]file-ID
```

Level-1 FASTCNFG File Syntax

In a level-1 FASTCNFG file you can:

- Redirect FASTP in a spooler subsystem to a subordinate FASTCNFG file (this FASTCNFG file would be a level-2 file):

```
SPOOLER [\node.]$supervisor-name
FASTCNFG = [\node.][$volume.][subvolume.]file-ID
```

- Specify device attributes for specific devices:

```
DEVICE [\node.]$device-name[. #qualifier]
attribute = attribute-value
[,attribute = attribute-value]...
```

Level-2 FASTCNFG File Syntax

In a level-2 FASTCNFG file you can:

- Redirect FASTP to a subordinate FASTCNFG file (this would be a level-3 file):

```
PROCESS $process-name
FASTCNFG = [\node.][$volume.][subvolume.]file-ID
```

- Specify device attributes for specific devices:

```
DEVICE [\node.]$device-name[. #qualifier]
attribute = attribute-value
[,attribute = attribute-value]...
```

Level-3 FASTCNFG File Syntax

In a level-3 FASTCNFG file you can only specify device attributes:

```
DEVICE [\node.]$device-name[. #qualifier]
attribute = attribute-value
[,attribute = attribute-value]...
```

EOLWRITE

FASTP normally attempts to buffer output, except in some cases involving parallel printers.

`EOLWRITE = YES | NO`

YES forces normally buffered output to be written to the device a line at a time.

The default value is NO.

ERRORRECOVERY

FASTP normally recovers from nonfatal errors during printing by restarting the entire job output.

`ERRORRECOVERY = YES | NO`

YES forces FASTP to recover from non-fatal errors by simply reissuing the last input/output (I/O) request. This is the same function that is available in PSPOOLB print processed when the PARM bit 5 is set for a device.

Considerations

ERRORRECOVERY is only recommended when the device involved can properly process a resend of the last I/O operation. Otherwise it can duplicate or corrupt output within the page where the error occurred.

TIMEOUT

The TIMEOUT attribute sets or overrides the wait time value established for I/O operations to a device. The value given is in one-hundredth (.01) of a second increments.

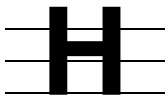
The default wait time for device I/O operations depends on the device type and its driver process. For devices connected using direct serial or parallel channels, FASTP waits indefinitely. It relies on the EIOTIMEOUT attribute configured for and applied by the device driver. For SNA connections, FASTP uses a default timeout of two minutes. For CONTROL 11 operations, FASTP uses a default timeout of one minute.

G Spooler Limits

This appendix describes the programmatic limits of the Spooler and Spooler Plus subsystem components.

Table G-1. Spooler Limits

Component	D46 and D48 Spooler	D41 Spooler	D41 Spooler With Spooler Plus	D40 Spooler
Jobs	8191	8191	65534	8191
Devices	4095	4095	4095	511
Locations	4095	4095	32767	4095
Print processes	45	45	255	45
Collector processes	16	16	31	16
Collector openers	1024	256	1024	256
Font jobs	255	255	2047	255
Batch jobs	4095	4095	32765	4095
Supervisor openers	1024	1024	1024	70
Supervisor message queue	1500	1500	1024	641



Migration Considerations for Pre-D41 Spooler Versions

The D41, D46, and D48 releases of the Spooler subsystem present many enhancements and new features to support improved operational management. This appendix describes D-series system interoperability considerations and explains the major enhancements and changes in the D41, D46, and D48 release of the Spooler subsystem. For information on new enhancements for D46 and D48, see [Extended Memory Changes for D46 and D48 Versions of the Spooler](#) on page H-12.

Certain, and essential, migration and installation considerations exist for pre-D41 Spooler product versions. Before upgrading to the newest version of Spooler, read this Appendix and especially [Migrating Files From a Pre-D41 Product Version of Spooler](#) on page H-2.

Pre-D41 Interoperability Considerations

Pre-D41, D41, D46 and D48 releases of the Spooler components interoperate without differences except as described below.

Pre-D41 Spoolcom and the D41,D46,D48 Spooler

- The COLLECT LISTOPENS command fails because it tries to get information directly from the collector process.
- The COLLECT STATUS DETAIL command now displays the larger number of units possible for a collector. Earlier Spoolcom utilities will display this data, as well as the percentage full value for a collector, inaccurately.
- The SPOOLER ERRLOG command fails; it is not supported.

Pre-D41 Peruse, D41, and the D46 and D48 Spooler Products

The JOB LIST command fails because the pre-D41 library procedures cannot interpret the D41 collection file format.

Pre-D41 Spooler, D41, and D46 and 48 Spoolcom Utilities

The COLLECT LISTOPENS command fails because it tries to get information from the supervisor process.

New Control File

A new format SPL3 file is introduced in the D46 and later D41 (ADI, ADM, and later) product versions of Spooler. The current Spooler product versions no longer use the SPL4 control file. In the SPL3 file, all information for jobs, batches, names, and attributes of other Spooler objects are integrated and retained in data structures. Thus, a warmstart from older Spooler versions requires both the SPL3 and SPL4 files and converts these files into the new SPL3 file.

Many developers and support analysts use the word “control version” to correspond with the numbering of Spooler control files. The current Spooler control version, Version 5 (D41/D46/D48 Spooler), corresponds to the current control file format, SPL3.

Before installing the D- or G-series operating system that introduces the new format SPL3 file, back up the Spooler control file in case you have to fall back. As mentioned earlier, and in [Section 1, Spooler Quick Start](#), when you warm start the Spooler, the control files are automatically converted to the new control file format.

Migrating Files From a Pre-D41 Product Version of Spooler

From a pre-D41 product version of Spooler, perform the following procedures to migrate to a spooler product version that is either T9101D41, D46, or D48:

1. Review the T9101D41 SOFTDOC that describes a workaround in case an old job is deleted after the new spooler product version is installed. See also [Section 3, Peruse "SPOOLCOM"](#) new options for JOB command
2. Use DSM/SCM to install the spooler software from the Site Update Tape (SUT).
3. Create new collector data file[s] that will be used by collectors after warm starting. Refer to [Section 4, Spoolcom](#) under JOB command for information about characteristics of collectors and their data files. See also [COLLECT Subcommand](#) on page H-9.
4. Back up the Spooler control and data files in case you have to fall back to your previous version of Spooler. For backup procedures, see [Step 3: Back up the Spooler Control and Data Files](#) on page 1-10. For migration information, see [Older-Version Collector Files Required to Access Older-Version Jobs](#) on page H-6I Control and Data Files.
5. Warm start the spooler. See [Section 1, Spooler Quick Start, Warm Starting a Spooler](#) on page 1-4.

Note. Do not use the SPOOLCOM;SPOOLER,START command at this time.

6. Change the DATA attribute of the existing collector[s] to point to the data file[s] created in step 3.

```
) COLLECT $S, DATA $SYSTEM.SPOOLER.NEWDATA
```

7. Start the spooler:

```
) SPOOLER, START
```

8. Use PERUSE and SPOOLCOM to manage jobs still residing in the pre-D41 collector files as well as the jobs being added to the new collector files.

D41 spooler Peruse commands have remained the same and can be run against jobs resident in old or new format collector files. The D41, D46, and D48 SPOOLCOM JOB qualifier and subcommand can also be used to manage the jobs in the old collector data file until such time as all jobs have been deleted, archived or printed; at this time the old data file can be deleted.

a. To change the location of a job using PERUSE:

```
_J1;LOC #NEW.LOC
```

b. To move jobs that have a certain LOC from a pre-D41 data file to a new data file, use SPOOLCOM.

```
)JOB (DATAFILE $SYSTEM.SPOOLER.OLDDATA, LOC  
#JOB) ,HOLD, COLLECT $$
```

```
)JOB 2 TRANSFERRED SUCCESSFULLY
```

c. If disk space is an issue, or you do not wish to deal with multiple data files, move all the jobs at the same time.

9. Once the jobs have been successfully moved to the new data file, you can delete the old file. The jobs are moved by passing them from the old data file to a collector that uses a data file created in step [3](#). The command is similar to this:

```
)JOB (DATAFILE $SYSTEM.SPOOLER.OLDDATA) ,HOLD, COLLECT $$
```

```
JOB 1 TRANSFERRED SUCCESSFULLY
```

```
JOB 2 TRANSFERRED SUCCESSFULLY
```

```
JOB 3 TRANSFERRED SUCCESSFULLY
```

```
JOB 4 TRANSFERRED SUCCESSFULLY
```

```
JOB 5 TRANSFERRED SUCCESSFULLY
```

D41, D46, D48 Common Enhancements and Changes

Compared to older product versions of the Spooler subsystem, the D41 release of the Spooler subsystem includes the following enhancements and changes; these enhancements and changes are included in the D46 and D48 versions of the Spooler:

- Increased object and resource limits
- New collector file requirements
- New collector data file requirements
- Improved collector performance
- New collector data file UNIT and DP2 BUFFERSIZE settings
- Older-version collector files required for older-version jobs
- More collector openers per collector
- Support for warmstarting from older-version spooler control files
- Changes to Spoolcom commands
- Changes to Spooler library procedures and structures
- New versions of print processes
- Increased extended memory
- New Event Message Service (EMS) event messages
- Increased support for Failure Data System (TFDS)
- Reduced support for PSPOOL and PSPOOLB print processes

These enhancements and changes are described in detail in the remainder of this appendix.

Increased Object and Resource Limits

[Table H-1](#) shows the object and resource limits that have changed in the D41 release of the Spooler subsystem.

Table H-1. D40 Spooler and D41 Spooler Limits Comparison

Object or Resource	D40 Spooler	D41 Spooler
Collector buffer size	1K	2K or a multiple of 2K
Maximum collector openers	256	1024
Maximum supervisor openers	70	1024
Units per collector file	8K	Limited only by file size

New Collector File Requirements

The format of the collector data file has been significantly altered. The new file format eliminates the space allocation constraints of previous collector designs and results in more efficient collection. You must create new collector data files in order to use the D41 or D46, or D48 Spooler product.

Although older versions of spooler collection files are used to access jobs present at the time of the warmstart, new jobs *cannot* be collected onto these files.

New Collector Data File Capacity Requirements

The number of units in a new collection file is no longer constrained by the spooler. Instead, it is limited solely by the amount of space allocated on the disk. In addition, unit size no longer needs to be inflated to allow use of larger collection files.

Improved Collector Performance

The elapsed time to collect jobs can be significantly shorter than in older-version spoolers when the data file `BUFFERSIZE` and collector `UNIT` attributes are set to allow larger buffers to be used by the collector.

You can obtain up to a 400 percent reduction in the elapsed time of collecting a job (particularly larger jobs) if the collector can use 4K buffers to do the collection I/O.

Older-version spooler collectors, whose jobs could be no greater than 8191 units, took advantage of this limit and retained in memory all accounting for the job's units allocated on disk. Because there is no longer a limit to the number of units in a job, the new collector uses an indexing scheme that is extensible. To minimize index overhead, the D41, D46, and D48 collector has been designed to promote contiguously allocated units and coalesce such allocation into a single index entry. This scheme minimizes the number of index records required for a job.

In cases where a job contains 1000 units or more and these units are largely allocated discontinuously on the disk, overhead will be greater than in older-version spoolers. As a result, the following operations may take a few seconds longer:

- Spoolcom `DEV SKIP` or `SKIPTO` commands
- `PERUSE LIST` specific-page command

More detailed analysis of the relative performance for these commands is available upon request from HP.

New Collector Data File `UNIT` and `DP2 BUFFERSIZE` Settings

As in older-version spoolers, the collector `UNIT` attribute is the amount of disk space allocated for a job each time space is required. The value entered is expressed in number of blocks of 1K bytes (for example, `UNIT 3` is equal to 3K bytes). Units contain data for only one job. Unused space in a unit is unavailable for other jobs. Once a collector data file has been initialized, the `UNIT` attribute cannot be altered. The minimum `UNIT` value is 2 (2K bytes).

The D41 collector buffers write operations, if possible, to match the collector data file's `DP2 BUFFERSIZE` attribute. This attribute is established at file creation (but can be altered later). Valid entries for collector use are 2048 or 4096 bytes or a multiple of 2048 bytes; a setting of 512 bytes is returned with an error message. You should set

the DP2 BUFFERSIZE to 4096 or to the minimum value that can contain the average size job collected.

A collector configuration is valid only if the unit size is a whole multiple of BUFFERSIZE. BUFFERSIZE must be at least 2 K bytes or a multiple of 2 K bytes. This is illustrated in [Table H-2](#).

Table H-2. Rcommended BUFFERSIZE Settings

DP2 BUFFER-SIZE(K)	Unit Size	Comment/Recommendation
2	Even	All even values of unit size are valid. For best performance, set BUFFERSIZE to 4K if unit size is also a whole multiple of 4.
4	2	This is valid. BUFFERSIZE could be set to 2 with no effect on performance. Consider increasing the unit size to 4 for best performance.
4	$n*4$	Valid for all values of n .
4	Other	All other values of unit size (for example, 5 or 6) are invalid.

D48 New Product Version of Spooler

This D48 product version includes the following files:

- EMS Template file ZSPLTMPL
- EMS Token definition files ZSPLC, ZSPLCOB, ZSPLDDL, ZSPLPAS, ZSPLTACL, and ZSPLTAL
- EMS Template source file SSPLTMPL
- ZSPLSEGF Segment file that contains TACL default segment

This product version of Spooler uses the control file version 5 and the current control file format, SPL 3.

Although this product version is similar in structure to D41 and D46 product versions, before upgrading to a new release or applying any new software product version (SPR), always back up the control files before loading the new release or applying the SPRs.

Older-Version Collector Files Required to Access Older-Version Jobs

The collector data files of the older-version spooler used in the warm start of a D41 spooler *must* be retained (not moved or renamed) if they contain jobs. When retained, these jobs are accessible to users immediately after warm start and may be perused, printed, deleted or moved to a new collector.

When all jobs in an older-version collector file have been deleted or moved, the file is no longer needed by the spooler. Older-version collector files used in this way are not altered in any way by the D41 spooler. JOB DELETE and MOVE operations are processed by deleting or altering the job information retained by the supervisor. Once a JOB DELETE command has been performed on one of these jobs, it can no longer be accessed, even though it has physically been retained in the old file.

More Collector Openers for Each Collector

D41, D46, and D48 collectors support 1024 concurrent openers. This support may allow the use of fewer collectors in some environments. Performance of a collector is primarily limited by disk access capacity.

Warmstart From Older-Version Spooler Files

Warm start from older-version spooler control files (file codes 124 through 128) is supported. The warm start operation is identical to that provided by previous spoolers except that access to existing jobs is provided by retaining access to the existing older-format collector files where they reside. These existing jobs are not moved or converted to the new collection file format during the warm start operation; instead, you must establish your own policies and procedures for the disposition of these jobs.

In some environments, existing jobs may be transient and will be deleted during the normal course of operation. In these cases, the command to select jobs residing in a specific data file can be used to identify when an older collector file can be removed from the system. In other cases, you may elect to move jobs immediately after warm start to free disk resources and to take advantage of new collector file settings.

Changes to Spoolcom Commands

This subsection describes changes to Spoolcom commands.

ERRLOG Subcommand

The ERRLOG subcommand, which was available to set the destination process or file for supervisor Event Message Service (EMS) messages, has been removed from the SPOOLER command. This subcommand in older versions of Spoolcom will also not function with the D41 spooler.

The destination process name for EMS messages from all spooler components is now established with a DEFINE provided by the process that runs SPOOL. The define is established for a particular EMS collector process with the following command:

```
ADD DEFINE =_EMS_COLLECTOR, FILE $process-name
```

If no define exists, the EMS messages will be directed to \$0.

COLLECT STATUS Command Display

The PERCENT FULL and %FULL values reported by the COLLECT STATUS command are now accurate to within plus or minus one percent.

Previously, the values retained in the supervisor for reporting were updated only at times when the collector completed a job or started a new job. Now, any time a value changes by 0.5 percent from the last time it was reported, the collector updates the supervisor with the new value. There is no performance impact as a result of this change.

COLLECT STATUS DETAIL Command Display

It is now possible to create and use collector data files that exceed 64K units. The COLLECT STATUS DETAIL command displays the larger number of units possible for a collector. Earlier Spoolcom utilities will display this data, as well as the percentage full value for a collector, inaccurately.

COLLECT LISTOPENS Command

The COLLECT LISTOPENS command has been redesigned. Spoolcom no longer gets the information for this command directly from the collector process. Earlier versions of Spoolcom can not be used to issue this command for D41 collectors, nor can the D41, D46, or D48 Spoolcom utility be used to issue this command to pre-D41 collectors.

DATAFILE Subcommand

A new JOB selection qualifier, DATAFILE, can be used with the JOB command. DATAFILE can be used by itself or in combination with other qualifiers to select jobs on which subcommands will be performed. If this qualifier is used, only jobs which reside in the specified collector data file are included in the selection. If the volume and subvolume portion of the data file name are omitted, the default volume and subvolume are used. The data file name must be specified and has no default.

[Example H-1](#) shows the SPOOLCOM JOB,STATUS command to status all jobs in all datafiles.

Example H-1. JOB Command to Status All Jobs in All Datafiles

) JOB

JOB	BATCH	STA	FLAGS	OWNER	TIME	COPY	PAGE	REPORT		LOCATION
1		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOBA
2		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB B
3		HLD	4	255,255	09:48	1	1	SUPER	SUPER	#JOB C
4		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB D
5		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB E
6		RDY	4	255,255	10:01	1	1	SUPER	SUPER	#D41.JOB

[Example H-2](#) shows the SPOOLCOM JOB, STATUS command to status all jobs in a particular data file, either in an old or new format. These jobs are from the datafile of a pre-D41 product version of Spooler.

Example H-2. JOB Command to Status all Jobs in a Particular Datafile

) JOB (DATAFILE \$SYSTEM.SPOOLER.DATA) , STATUS

JOB	BATCH	STA	FLAGS	OWNER	TIME	COPY	PAGE	REPORT		LOCATION
1		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOBA
2		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB B
3		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB C
4		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB D
5		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB E

[Example H-3](#) shows STATUS jobs of a particular data file and location.

Example H-3. JOB Command to Status a Particular Datafile and Location

) JOB (DATAFILE \$SYSTEM.SPOOLER.DATA, LOC #JOB C) , STATUS

JOB	BATCH	STA	FLAGS	OWNER	TIME	COPY	PAGE	REPORT		LOCATION
3		RDY	4	255,255	09:48	1	1	SUPER	SUPER	#JOB C

COLLECT Subcommand

A new subcommand, COLLECT, can be used with the JOB command to move selected job(s) from their current collector data file to the specified collector process.

All job parameters are retained and the job is rapidly recollected into the new collector data file.

For the COLLECT subcommand to function on a job, the job must be in the HOLD state. If the job has a current, active collector, it is deleted from the current collector after the transfer has completed successfully. If the job does not belong to a collector that is active (for example, jobs warm-started from an older-version spooler), the job is not physically deleted from the current data file, but is no longer accessible from that file after the transfer is successful. If a transfer fails, the job remains on the current data file and/or collector. Successful transfer of a job is reported by the following message issued to the Spoolcom user:

```
JOB nnnn TRANSFERRED TO $process-name SUCCESSFULLY
```

Messages associated with unsuccessful transfers are as follows:

```
JOB nnnn TRANSFER TO $process-name FAILED, ERROR nnn
COLLECTOR $process-name OPEN ERROR nnn
```

[Example H-4](#) shows transferring jobs from a pre-D41 format collector file to a collector with a D41, D46, or D48 format collector file:

Example H-4. Transferring Jobs

```
)JOB (DATAFILE $SYSTEM.SYSTEM.DATA),HOLD,COLLECT $S
JOB 1      TRANSFERRED SUCCESSFULLY
JOB 2      TRANSFERRED SUCCESSFULLY
JOB 3      TRANSFERRED SUCCESSFULLY
JOB 4      TRANSFERRED SUCCESSFULLY
JOB 5      TRANSFERRED SUCCESSFULLY
```

Note. You can use the COLLECT subcommand to transfer jobs between collectors that belong to the same spooler supervisor process.

Changes to Spooler Library Procedures and Structures

Some Spooler library procedures (for example, PRINTREAD) have been significantly changed internally. Callers of these procedures provide a buffer of a specified size for these procedures to use. The contents of the buffer are accessible to applications, but should not be accessed and used directly. The format and content of this buffer has been changed.

Applications that use the procedures to obtain information from the buffer will not be affected by the internal changes. However, if an application directly accesses the contents of the buffer, it will probably not function correctly with the D41 spooler library.

The following three system library spooling procedures are no longer applicable for access to jobs that reside in the D41 product version of Spooler (first available in the D44 and G03 system releases):

PRINTCOMPLETE
PRINTSTART
SPOOLERREQUEST

These three procedures can continue to be used, however, if the application accesses jobs only in the form of filecode 129 job files. Customers wishing to access jobs in the D41 product version of Spooler must now employ the “2” versions of these procedures:

PRINTCOMPLETE2
PRINTSTART2
SPOOLERREQUEST2

See Hotstuff HS02226 for detailed information.

To help determine if a user application is using these library spooling procedures, you can use a specific TACL MACRO. The TACL MACRO searches the binder region of specified filecode 100 files for any of the three procedure names. The macro can be found attached to Hotstuff HS02226.

Unsupported Procedures

You cannot use the PRINTCOMPLETE, PRINTSTART, and SPOOLERREQUEST procedures to access spooler jobs that reside in the D41 and later releases of the Spooler subsystem. You can continue to use these procedures to access filecode 129 job files, however. To access spooler jobs in the D41 and later releases of the Spooler subsystem, you must use the PRINTCOMPLETE2, PRINTSTART2, and SPOOLERREQUEST2 procedures.

New Versions of Print Processes Required

The following print processes have been altered in order to operate with the D41, D46, or D48 spooler:

- NETWARE LAN PRINT
- XNS LAN PRINT
- TCP/IP LAN PRINT
- X.25 WAN PRINT
- FASTPLW

If you plan to use any of these products, and have not yet upgraded to at least a D44 or G03 NonStop Kernel operating system, you must install the correct SPR for each product. Always check with IPM Scout for the latest SPRs.

Increased Extended Memory

The spooler supervisor process now uses approximately 30 megabytes (Mbytes) of flat extended segment space. Each spooler collector process now uses from 2 to 5 Mbytes of flat extended segment space depending on the buffer size used. The collector attempts to match the DP2 BUFFERSIZE.

New EMS Event Messages

New tokenized event messages are now provided to replace many of the Spooler-generated Event Management Service (EMS) type 512 messages. The new event messages are described in the *Operator Messages Manual*.

Increased Support for Failure Data System (TFDS)

TFDS is now used for all fatal error reporting and processing. Improved diagnosis information has also been provided as a by-product of introducing TFDS to the spooler.

Reduced Support for PSPOOL and PSPOOLB Print Processes

FASTP and the available value-added products for specific network printing (TCP/IP LAN Print and LAN Print for Netware, for example) are now the only print processes documented and fully supported. However, D41 versions of PSPOOL and PSPOOLB are provided to minimize migration impact. The D41 versions of PSPOOL and PSPOOLB accept the D41 spooler library version but are otherwise unchanged from previous versions. You cannot use pre-D41 versions of PSPOOL and PSPOOLB.

Extended Memory Changes for D46 and D48 Versions of the Spooler

The D46 and D48 versions of the Spooler subsystem include the enhancements and changes featured in the D41 Spooler. In addition, the D46 and D48 versions of the Spooler use extended memory differently than the D41 Spooler version. For the D46 and D48 versions of the Spooler you may need to increase the extended memory on your system because the D46 and D48 versions use more memory than earlier Spooler versions. Check with IPM Scout or ExpressNotice for the latest SPR.

The following is a worksheet for calculating extended memory space.

D46 and D48 Spooler memory usage for internal tables:

```
Spooler Supervisor Memory usage for its major tables =
  STRUCT JOB^STRUCT          297 bytes * MAX number of jobs
+ STRUCT DEV^STRUCT          112 bytes * MAX number of devices
+ STRUCT CDEV^STRUCT         62 bytes * MAX number of devices
+ STRUCT COL^STRUCT          132 bytes * MAX number of collectors
+ STRUCT CCOL^STRUCT         136 bytes * MAX number of collectors
+ STRUCT PRT^STRUCT          85 bytes * MAX number of print processes
+ STRUCT CPRT^STRUCT         44 bytes * MAX number of print processes
+ STRUCT FONT^STRUCT         40 bytes * MAX number of font jobs
+ STRUCT BATCH^STRUCT        146 bytes * MAX number of batches
+ STRUCT NODE_STRUCT_DEF     50 bytes * MAX number of queue nodes
+ STRUCT LOC^STRUCT          86 bytes * MAX number of locations
+ STRUCT USER^STRUCT         298 bytes * 1024
```

Per Supervisor:----- Maximum configurations -----

	SPOOLER Max #	SPLR PLUS Max #	SPOOLER Bytes	SPLR PLUS Bytes	
297 * JOBS	8191	85534	2432727	19463598	(19.4 Mb)
+ 174 * DEVICES	4095	4095	712530	712530	(0.7 Mb)
+ 268 * COLLECTORS	16	31	4288	8308	
+ 129 * PRINT PROCS	45	255	5805	32895	
+ 40 * FONT	255	2047	10200	81880	
+ 146 * BATCHES	4095	32767	597870	4783982	(4.7 Mb)
+ 86 * LOCATIONS	4095	32767	352170	2817962	(2.8 Mb)
+ 50 * QUEUE NODES	131086	131086	6554300	6554300	(6.5 Mb)
+ 298 * OPENERS	1024	1024	305152	305152	(0.3 Mb)
			10975042	34760607	(34.7 Mb)
		(Incl Backup)	21950084	69521214	(69.5 Mb)

Per Collector:

	1K Buffer	4K Buffer	
1024 * (1+BUFFERSIZE) * OPENERS (1024)	2097152	5242880	(5.2 Mb)
(Incl backup)	4194304	10485760	(10.5 Mb)

Sample configurations and the memory required for the tables are below.

Spooler Maximum: 32K queue nodes, 1 collector:

2 Supervisors	5435526	(5.4 Mb)
2 Collectors	4194304	(4.2 Mb)

	9629830	(10 Mb)

Spooler Maximum: default queue nodes, 3 collectors:

2 Supervisors	10351426	(10 Mb)
6 Collectors	31457280	(31 Mb)

	42160132	(42.2 Mb)

Spooler Plus Maximum: default queue nodes, 3 collectors:

2 Supervisors	10351426	(10.3 Mb)
6 Collectors	31457280	(31.5 Mb)

	41808706	(41.8 Mb)

Spooler Plus: default queue nodes, 31 collectors:

2 Supervisors	68971940	(68 Mb)
62 Collectors	325058560	(325 Mb)

	394030500	(394 Mb)

Index

Numbers

129 files

See Spooler job files

A

Active state (collector), explained [4-16](#)

Active state (spooler), explained [4-67](#)

ALIGN subcommand, SPOOLCOM DEV command [4-20](#)

ALL option

PERUSE HELP command [3-23](#)

PERUSE LIST command [3-34](#)

Archiving example [3-36](#)

ASCII compression example [3-37](#)

ASSIGN PRINTFILE command (RPSETUP) [6-11](#)

Authorization

FONT utility [5-4](#)

B

Backing up all files [1-11](#)

Backing up control and data files [1-10](#)

Backup CPU, for interspooler threads [6-11](#)

BACKUP subcommand

SPOOLCOM COLLECT command [4-12](#)

SPOOLCOM PRINT command [4-63](#)

BACKUP subcommand, SPOOLCOM COLLECT [2-5](#)

BATCH command

PERUSE [3-11](#)

SPOOLCOM [4-7/4-11](#)

Batch jobs, managing [2-18/2-22](#)

Batch jobs, number of [A-3](#)

Batch number [2-9](#)

BOTH subcommand, PERUSE FIND command [3-19](#)

BREAK key, while PERUSE is listing a job [3-35](#)

BROADCAST subcommand

SPOOLCOM LOC command [4-58](#)

Busy state (device), explained [4-35](#)

C

Character translation [5-11](#)

CHARMAP subcommand, SPOOLCOM DEV command [4-20](#)

CLEAR subcommand, SPOOLCOM DEV command [4-21](#)

Coldstart parameters

SPOOL program [A-2](#)

Coldstarting the spooler

SPOOL program [A-3](#)

Coldstart, definition of [1-8](#)

COLLECT command (SPOOLCOM) [4-12/4-18](#)

COLLECT qualifier, SPOOLCOM JOB command [4-48](#)

COLLECT subcommand [H-9](#)

SPOOLCOM FONT command [4-43](#)

Collector [2-5/2-7](#)

attributes [2-5](#)

large and small jobs [2-7](#)

optimizing performance [2-5](#)

spooler [4-12/4-18](#)

states, explained [4-16](#)

unit size for [2-7](#)

Collectors, number of [A-2](#)

Command summary

PERUSE [3-9](#)

SPOOLCOM [4-5](#)

COMMENT command (SPOOLCOM) [4-19](#)

Comments, in script file [5-16](#)

Configuration for device timeout attribute [4-30](#)

Configuration parameters (RPSETUP) [6-8](#)

CONTROL directive (FONT utility) [5-9](#)

Control file name

SPOOL program [A-1](#)

Control file, new for D41 and D46 [H-2](#)

CONTROLBUF directive (FONT utility) [5-9/5-12](#)

COPIES command (PERUSE) [3-12](#)

COPIES subcommand

SPOOLCOM BATCH command [4-7](#)

SPOOLCOM JOB command [4-50](#)

CPU subcommand

SPOOLCOM COLLECT
command [4-12](#)

SPOOLCOM PRINT command [4-63](#)

CPU subcommand, SPOOLCOM
COLLECT [2-5](#)

Cross-connected spoolers [6-1](#)

D

Data file, collector [4-12](#), [4-14](#)

Data stream compatibility (DSC)
mode [4-22](#)

DATA subcommand

SPOOLCOM COLLECT
command [4-12](#)

DATA subcommand, SPOOLCOM
COLLECT [2-5](#)

DATAFILE qualifier, SPOOLCOM JOB
command [4-48](#)

DATAFILE subcommand [H-8](#)

DATE qualifier, SPOOLCOM FONT
command [4-43](#)

DATE qualifier, SPOOLCOM JOB
command [4-48](#)

DEBUG subcommand

SPOOLCOM PRINT command [4-63](#)

DEBUG subcommand, SPOOLCOM PRINT
command [4-63](#)

DEL command (PERUSE) [3-13/3-14](#)

DELETE subcommand (SPOOLCOM)

BATCH command [4-7](#)

COLLECT command [4-12](#)

DELETE subcommand
(SPOOLCOM) (continued)

DEV command [4-21](#)

FONT command [4-42](#)

PRINT command [4-64](#)

Destination name [2-15/2-17](#)

JOB command (PERUSE) [3-28](#)

Destination, logical and physical [2-15](#)

DETAIL subcommand, SPOOLCOM FONT
command [4-44](#)

DEV command

PERUSE [3-15](#)

SPOOLCOM [4-20/4-38](#)

DEV subcommand

SPOOLCOM LOC command [4-58](#)

Deverror state (device), explained [4-35](#)

Device [2-10/2-14](#)

connect to a location [4-58](#)

controlling (SPOOLCOM) [4-20/4-38](#)

displaying status (PERUSE) [3-15](#)

displaying status
(SPOOLCOM) [4-20/4-38](#)

interspooler threads

assigning header [6-12](#)

assigning ownership [6-11](#)

assigning selection algorithm [6-12](#)

maximum line width [6-15](#)

parameter option [6-13](#)

restart interval [6-14](#)

truncation options [6-14](#)

number of [A-2](#)

special form or special paper [3-21](#)

states, explained [4-35](#)

Device-queue-nodes, maximum
number [A-3](#)

DEVRESET subcommand

SPOOLCOM DEV command [4-21](#)

DEVTYPE subcommand

SPOOLCOM DEV command [4-21](#)

Dormant state (collector), explained [4-16](#)

Dormant state (spooler), explained [4-67](#)

Drain state (collector), explained [4-16](#)
 DRAIN subcommand [1-10](#)
 SPOOLCOM COLLECT
 command [4-13](#)
 SPOOLCOM DEV command [4-22](#)
 SPOOLCOM SPOOLER
 command [4-70](#)
 Draining the spooler [1-9](#)
 DUMP subcommand
 SPOOLCOM SPOOLER
 command [4-70](#)
 D-series system and the SPOOLCOM
 command [4-3](#)

E

ENDFF subcommand, SPOOLCOM DEV
 command [4-22](#)
 END-FONT directive [5-6](#)
 END-LOCATIONS directive [5-15](#)
 Error log file name
 SPOOL program [A-1](#)
 Error messages
 FONT utility [D-1/D-21](#)
 PERUSE [B-1/B-10](#)
 RPSETUP utility [E-1/E-3](#)
 SPOOLCOM [C-1/C-9](#)
 Error state (collector), explained [4-16](#)
 Error state (spooler), explained [4-67](#)
 EXCLUSIVE subcommand
 SPOOLCOM DEV command [4-23](#)
 Execution priority
 SPOOL program [A-1](#)
 EXIT command
 PERUSE [3-16](#)
 SPOOLCOM [4-40](#)
 Extended memory [H-12](#)
 description of [H-12](#)
 worksheet for calculations [H-12](#)

F

FASTP
 tracing [F-2](#)
 FASTP (print process)
 configuring [F-1](#)
 configuring with SPOOLCOM [4-22](#),
 [4-24](#), [4-25](#), [4-29](#)
 CONTROLBUF operations [5-10](#)
 default operation [6-13](#)
 FONT jobs [5-10](#)
 FC command
 PERUSE [3-17](#)
 SPOOLCOM [4-41](#)
 FIFO option, for interspooler threads [6-12](#)
 FIFO subcommand, SPOOLCOM DEV
 command [4-23](#)
 FILE subcommand
 SPOOLCOM COLLECT
 command [4-13](#)
 SPOOLCOM PRINT command [4-64](#)
 FILE subcommand, SPOOLCOM
 COLLECT [2-5](#)
 FILES command (PERUSE) [3-18](#)
 FIND command (PERUSE) [3-19/3-20](#)
 FONT command (FONT utility) [5-3](#)
 FONT command (SPOOLCOM) [4-42/4-46](#)
 Font data, normal [5-7/5-8](#)
 FONT directive (FONT utility) [5-7](#)
 Font job
 characteristics [5-4](#)
 copying data [5-14](#)
 displaying modifiers [5-4](#)
 FASTP (print process) [5-12](#)
 FONT command (SPOOLCOM) [4-42](#)
 FONT utility [5-1](#)
 Font job sections, number of [5-12](#)
 Font jobs, number of [A-3](#)

FONT utility [5-1/5-17](#)

- CONTROL directive [5-9](#)
- CONTROLBUF directive [5-9/5-12](#)
- END-FONT directive [5-6](#)
- END-LOCATIONS directive [5-15](#)
- error messages [D-1/D-21](#)
- FONT directive [5-7](#)
- normal data [5-7/5-8](#)
- SETMODE directive [5-13](#)
- SOURCE directive [5-14](#)
- START-FONT directive [5-6](#)
- START-LOCATIONS directive [5-15](#)

FORM command (PERUSE) [3-21/3-22](#)

Form name

- device [2-12](#)
- job [2-10](#)

FORM subcommand

- SPOOLCOM BATCH command [4-7](#)
- SPOOLCOM DEV command [4-23](#)
- SPOOLCOM FONT command [4-43](#)
- SPOOLCOM JOB command [4-49](#), [4-50](#)

Function keys, displaying a job (PERUSE) [3-8](#)

G

Group [2-15/2-17](#)

Group manager access

- JOB command (PERUSE) [3-29](#)
- MGRACCESS subcommand [4-70](#)
- SHOWGROUP command (PERUSE) [3-47](#)
- STATUS command (PERUSE) [3-51](#)

Group name

- JOB command (PERUSE) [3-28](#)

H

Header format, for interspooler threads [6-12](#)

Header message, device [2-13](#)

HEADER subcommand, SPOOLCOM DEV command [4-23](#)

HELP command

- PERUSE [3-23](#)
- SPOOLCOM [4-47](#)

HOLD command (PERUSE) [3-24/3-25](#)

HOLD subcommand

- SPOOLCOM BATCH command [4-7](#)
- SPOOLCOM JOB command [4-50](#)

HOLDAFTER command (PERUSE) [3-26/3-27](#)

HOLDAFTER subcommand

- SPOOLCOM BATCH command [4-8](#)
- SPOOLCOM JOB command [4-50](#)

Hold-after-printing flag [3-26](#)

I

Interspooler threads

- assigning thread name [6-17](#)
- configuring number [6-15](#)
- illustrated [6-3](#)

J

Job

- altering column number (PERUSE) [3-50](#)
- altering location

 - PERUSE [3-38](#)
 - SPOOLCOM [4-50](#)

Job (continued)

altering number of columns
(PERUSE) [3-39](#)

altering number of copies
(PERUSE) [3-12](#)

altering owner of
PERUSE [3-41](#)
SPOOLCOM [4-51](#)

altering queue priority
PERUSE [3-44](#)
SPOOLCOM [4-51](#)

altering report name
PERUSE [3-46](#)
SPOOLCOM [4-51](#)

altering status of
(SPOOLCOM) [4-48/4-57](#)

attributes [2-9/2-10](#)

batch number [2-9](#)

copies [2-10](#)

deleting from spooler
PERUSE [3-13](#)
SPOOLCOM [4-50](#)

described [2-9/2-10](#)

displaying a line of (PERUSE) [3-19](#)

displaying pages of (PERUSE) [3-8](#),
[3-33](#)

displaying status of
PERUSE [3-28](#), [3-51](#)
SPOOLCOM [4-51](#)

form name [2-10](#)

linking and unlinking [2-21](#)

linking to current batch job
(PERUSE) [3-32](#)

location [2-10](#)

number [2-9](#)

obtaining status of
(SPOOLCOM) [4-48/4-57](#)

printing pages (PERUSE) [3-33](#)

priority [2-10](#)

Job (continued)

queue, displaying status
(PERUSE) [3-15](#)

report name [2-10](#)

setting current job (PERUSE) [3-28](#)

state [2-10/2-11](#)

stop printing current
(SPOOLCOM) [4-21](#)

unlinking from current batch job
(PERUSE) [3-53](#)

JOB command
PERUSE [3-28/3-31](#)
SHOWGROUP command
(PERUSE) [3-47](#)
SPOOLCOM [4-48/4-57](#)

Job number zero
align template [4-35](#)
corrupted job map [A-3](#)

JOB subcommand
SPOOLCOM DEV command [4-24](#)
SPOOLCOM FONT command [4-42](#)

JOBID job attribute [2-21](#)

Jobs, number of [A-2](#)

K

Kanji translation [4-20](#)

L

LINK command (PERUSE) [3-32](#)

LINK subcommand, SPOOLCOM BATCH
command [4-8](#)

Linked spoolers [2-7](#), [6-1](#)

LIST command (PERUSE) [3-33/3-37](#)
displaying a job [3-8](#)

LOC command
PERUSE [3-38](#)
SPOOLCOM [4-58/4-61](#)

LOC subcommand
 SPOOLCOM BATCH command [4-8](#)
 SPOOLCOM FONT command [4-43](#)
 SPOOLCOM JOB command [4-49](#),
 [4-50](#)

Local spooler, RPSETUP [6-6](#)

Location [2-10](#), [2-15/2-17](#)
 altering
 PERUSE [3-38](#)
 SPOOLCOM [4-50](#)
 sending process output to [2-17](#)

Locations section [5-15](#)

Locations, number of [A-2](#)

Log file name
 SPOOL program [A-1](#)

LUEOLVALUE subcommand, SPOOLCOM
 DEV command [4-24](#)

LUEOLWHEN subcommand, SPOOLCOM
 DEV command [4-25](#)

LUTOFVALUE subcommand, SPOOLCOM
 DEV command [4-25](#)

M

Manager access
 See Group manager access

MBCS [4-20](#)

Memory, extended [H-12](#)

MGRACCESS subcommand [3-47](#)
 SPOOLCOM SPOOLER
 command [4-70](#)

Migration
 spooling procedures [H-10](#)

Multibyte character set [4-20](#)

N

Network spooling
 linked spoolers [2-7](#), [6-1/6-5](#)
 SPOOLCOM command [4-3](#)

Normal font data [5-7/5-8](#)

NUMCOL command (PERUSE) [3-39](#)
 affects line displayed
 FIND command [3-20](#)
 LIST command [3-35](#)

O

Offline state (device), explained [4-35](#)

OPEN command
 PERUSE [3-40](#)
 SPOOLCOM [4-62](#)

OUT subcommand, SPOOLCOM FONT
 command [4-44](#)

OWNER command (PERUSE) [3-41](#)

OWNER subcommand
 SPOOLCOM BATCH command [4-8](#)
 SPOOLCOM FONT command [4-43](#),
 [4-49](#)
 SPOOLCOM JOB command [4-51](#)

Owner, for interspooler threads [6-11](#)

P

PAGE command
 PERUSE [3-42](#)

PAGES subcommand
 SPOOLCOM FONT command [4-44](#)
 SPOOLCOM JOB command [4-49](#)

PAGESIZE subcommand
 SPOOLCOM COLLECT
 command [4-13](#)

PAGESIZE subcommand, SPOOLCOM
 COLLECT [2-5](#)

PARAM BUPCPU command
 (RPSETUP) [6-11](#)

PARAM DEVEXCLUSIVE command
 (RPSETUP) [6-11](#)

PARAM DEVFIFO command
 (RPSETUP) [6-12](#)

PARAM DEVHEADER command
 (RPSETUP) [6-12](#)

PARAM DEVPARM command
 (RPSETUP) [6-13](#)

PARAM DEVRESTART command
(RPSETUP) [6-14](#)

PARAM DEVTRUNC command
(RPSETUP) [6-14](#)

PARAM DEVDWIDTH command
(RPSETUP) [6-15](#)

PARAM NUMTHREADS command
(RPSETUP) [6-15](#)

PARAM PRICPU command
(RPSETUP) [6-15](#)

PARAM PRINTNAME command
(RPSETUP) [6-16](#)

PARAM PRINTPARG command
(RPSETUP) [6-16](#)

PARAM PRIORITY command
(RPSETUP) [6-16](#)

PARAM THREADNAME command
(RPSETUP) [6-17](#)

Parameters, configuring with
RPSETUP [6-8/6-17](#)

PARM subcommand

- SPOOLCOM DEV command [4-25/4-27](#)
- SPOOLCOM PRINT command [4-64](#)

Pattern matching, SPOOLCOM REPORT
command [4-49](#)

PERUSE

- command summary [3-9](#)
- command to run [3-3](#)
- commands
 - BATCH [3-11](#)
 - COPIES [3-12](#)
 - DEL [3-13/3-14](#)
 - DEV [3-15](#)
 - EXIT [3-16](#)
 - FC [3-17](#)
 - FILES [3-18](#)
 - FIND [3-19/3-20](#)
 - FORM [3-21/3-22](#)
 - HELP [3-23](#)
 - HOLD [3-24/3-25](#)
 - HOLDAFTER [3-26/3-27](#)
 - JOB [3-28/3-31](#)

PERUSE, commands (continued)

- LINK [3-32](#)
- LIST [3-33/3-37](#)
- LOC [3-38](#)
- NUMCOL [3-39](#)
- OPEN [3-40](#)
- OWNER [3-41](#)
- PAGE [3-42](#)
- PRI [3-44](#)
- PURGE [3-45](#)
- REPORT [3-46](#)
- SHOWGROUP [3-47/3-48](#)
- SJFILES [3-49](#)
- STARTCOL [3-50](#)
- STATUS [3-51/3-52](#)
- UNLINK [3-53](#)
- VOLUME [3-54](#)

declaring the current job [3-7](#)

displaying a job [3-8](#)

error messages [B-1/B-10](#)

running interactively [3-4](#)

running noninteractively [3-4](#)

PERUSE and SPOOLCOM,
differences [2-9](#)

PREXLATE subcommand, SPOOLCOM
DEV command [4-27](#)

PRI command (PERUSE) [3-44](#)

PRI subcommand

- SPOOLCOM COLLECT
command [4-13](#)
- SPOOLCOM PRINT command [4-64](#)

PRI subcommand, SPOOLCOM
COLLECT [2-5](#)

PRINT command (SPOOLCOM) [4-63/4-69](#)

Print process [4-65](#)

Print processes

- number of [A-3](#)

Print Process, interspooler threads

- assigning primary CPU [6-15](#)

Print process, interspooler threads

- assigning backup CPU [6-11](#)
- assigning priority [6-16](#)
- assigning process name [6-16](#)
- assigning program file [6-11](#)
- assigning startup message [6-16](#)
- default operations [6-13](#)

Print process, spooler

- changing attributes of (SPOOLCOM) [4-65](#)
- changing status of (SPOOLCOM) [4-63/4-69](#)
- independent [4-64](#)
- obtaining status of (SPOOLCOM) [4-63/4-69](#)
- specifying attributes of (SPOOLCOM) [4-63/4-69](#)
- specifying the program [4-64](#)
- states [4-67](#)
- user-written [4-64](#)

Printer

- device types [4-21/4-22](#)

Printer, laser

- device configuration example [4-39](#)

Priority, job [2-10](#)

Priority, SPOOL program [A-1](#)

Procerror state (device), explained [4-35](#)

PROCESS subcommand, SPOOLCOM DEV command [4-27](#)

PURGE command (PERUSE) [3-45](#)

PURGEDATA [1-8](#)

Purging the datafile before coldstart [1-8](#)

Q

Queue, altering job position in (PERUSE) [3-44](#)

R

REBUILD option, SPOOL program [A-3](#)

Remote spooler [6-1/6-5](#)

Remote spooler, RPSETUP [6-6](#)

Remote system, viewing spooled job

- PERUSE [3-40](#)
- SPOOLCOM [4-62](#)
- SPOOLCOM command [4-3](#)

REPORT command (PERUSE) [3-46](#)

Report name [2-10](#)

REPORT subcommand

- SPOOLCOM BATCH command [4-8](#)
- SPOOLCOM FONT command [4-44](#)
- SPOOLCOM JOB command [4-49, 4-51](#)

Resetting, a device [4-21](#)

Restart interval, for interspooler threads [6-14](#)

RESTART subcommand, SPOOLCOM DEV command [4-27](#)

RETRY subcommand, SPOOLCOM DEV command [4-28](#)

Retryable operations

- number of retries [4-30](#)
- retry interval [4-30](#)

RETURN key, displaying a job (PERUSE) [3-8](#)

Routing structure [2-15/2-17](#)

RPSETUP utility [6-1/6-17](#)

- command to run [6-6](#)
- configuration parameters [6-8/6-17](#)
- error messages [E-1/E-3](#)

RPSetup utility

- description of [1-13](#)

Run options

- with FONT utility [5-3](#)
- with PERUSE command [3-3](#)
- with SPOOLCOM command [4-2](#)

S

Script file

- comments [5-16](#)
- example [5-17](#)
- syntax [5-5/5-17](#)

Security

- FONT utility [5-4](#)
- RPSETUP utility [6-7](#)
- spooler and individual users [2-22](#)

SELPRI subcommand

- SPOOLCOM BATCH command [4-8](#)
- SPOOLCOM JOB command [4-51](#)

SETMODE directive (FONT utility) [5-13](#)

SHOWGROUP command (PERUSE) [3-47/3-48](#)

SJFILES command (PERUSE) [3-49](#)

SKIP subcommand, SPOOLCOM DEV command [4-28](#)

SKIPTO subcommand, SPOOLCOM DEV command [4-28](#)

SNA character string (SCS) mode [4-22](#)

SOURCE directive (FONT utility) [5-14](#)

SPEED subcommand, SPOOLCOM DEV command [4-28](#)

SPL3 [H-2](#)

SPL3 file [H-2](#)

SPOOL DEFINE, sending process output to [2-18](#)

SPOOL program [A-1/A-4](#)

SPOOLCOM

command summary [4-5](#)

command to run [4-2](#)

commands

- BATCH [4-7/4-11](#)
- COLLECT [4-12/4-18](#)
- COMMENT [4-19](#)
- DEV [4-20/4-38](#)
- EXIT [4-40](#)
- FC [4-41](#)
- FONT [4-42/4-46](#)
- HELP [4-47](#)

SPOOLCOM commands (continued)

JOB [4-48/4-57](#)

LOC [4-58/4-61](#)

PRINT [4-63/4-69](#)

SPOOLER [4-70/4-71](#)

error messages [C-1/C-9](#)

opening spooled job [4-62](#)

SPOOLCOM LOC

BROADCAST subcommand [4-58](#)

DEV subcommand [4-58](#)

Spooler

collector, See Collector

components [2-3](#)

displaying component statistics [4-71](#)

displaying component status [4-71](#)

network configuration [2-7](#), [6-1/6-5](#)

routing structure, altering (SPOOLCOM) [4-58/4-61](#)

security [2-22](#), [4-2](#)

states, explained [4-67](#)

unit size [2-7](#)

Spooler configuration file, example of [1-2](#)

Spooler job

See Job

Spooler job files [3-8](#)

Spooler location [2-15/2-17](#)

sending process output to [2-17](#)

Spooler states, explained [4-67](#)

Spooler supervisor

remote

opening with PERUSE [3-40](#)

opening with SPOOLCOM [4-62](#)

running

PERUSE [3-3](#)

SPOOLCOM [4-2](#)

Spoolers, linked [2-7](#), [6-1](#)

START subcommand

SPOOLCOM BATCH command [4-8](#)

SPOOLCOM COLLECT command [4-13](#)

START subcommand (continued)
 SPOOLCOM DEV command [4-28](#)
 SPOOLCOM JOB command [4-51](#)
 SPOOLCOM PRINT command [4-64](#)
 SPOOLCOM SPOOLER
 command [4-70](#)

STARTCOL command (PERUSE) [3-50](#)
 affects line displayed
 FIND command [3-20](#)
 LIST command [3-35](#)

STARTFF subcommand, SPOOLCOM DEV
 command [4-29](#)

Starting the spooler
 SPOOL program [A-1](#)

START-FONT directive [5-6](#)

START-LOCATIONS directive [5-15](#)

STATE subcommand
 SPOOLCOM FONT command [4-44](#)
 SPOOLCOM JOB command [4-49](#)

States of spooler components
 collector [4-16](#)
 device [4-31](#)
 job [4-51](#)
 print process [4-67](#)

STATISTICS subcommand
 SPOOLCOM SPOOLER
 command [4-71](#)

STATUS command
 SHOWGROUP command
 (PERUSE) [3-47](#)

STATUS command (PERUSE) [3-51/3-52](#)

STATUS subcommand
 PERUSE JOB command [3-28](#)
 SPOOLCOM BATCH command [4-8](#)
 SPOOLCOM COLLECT
 command [4-13](#)
 SPOOLCOM DEV command [4-29](#)
 SPOOLCOM FONT command [4-44](#)
 SPOOLCOM JOB command [4-51](#)
 SPOOLCOM PRINT command [4-64](#)

STATUS subcommand (continued)
 SPOOLCOM SPOOLER
 command [4-71](#)

Supervisor name
 \$SPLS [A-1](#)

Supervisor process, identifying [1-11](#)

SUSPEND subcommand, SPOOLCOM
 DEV command [4-30](#)

Suspended state (device), explained [4-35](#)

T

TACL macro to start Spooler [1-2](#)

TACL STOP command [1-10](#)

TFORM [3-20](#), [3-42](#)

TIMEOUT
 number of retries [4-30](#)
 retry interval [4-30](#)

TIMEOUT subcommand, SPOOLCOM DEV
 command [4-30](#)

Top-of-form substitution [5-11](#)

TRUNC subcommand, SPOOLCOM DEV
 command [4-31](#)

Truncation option, for interspooler
 threads [6-14](#)

U

Unit size [2-7](#)

UNIT subcommand
 SPOOLCOM COLLECT
 command [4-14](#)

UNIT subcommand, SPOOLCOM
 COLLECT [2-5](#)

UNLINK command (PERUSE) [3-53](#)

UNLINK subcommand
 SPOOLCOM BATCH command [4-9](#)

V

Virtual font jobs [5-12](#)

VOLUME command (PERUSE) [3-54](#)

W

Waiting state (device), explained [4-35](#)

Warmstarting the spooler

 SPOOL program [A-3](#)

Warmstart, definition of [1-8](#)

WIDTH subcommand

 SPOOLCOM DEV command [4-31](#)

X

XREF subcommand

 SPOOLCOM DEV command [4-31](#)

 SPOOLCOM PRINT command [4-65](#)

Special Characters

\$SPLS (default spooler supervisor) [3-3](#),
[3-40](#), [4-2](#), [4-62](#), [5-3](#), [A-1](#)

\$SYSTEM.SYSTEM.CSPOOL [2-5](#), [4-13](#),
[4-18](#)

\$SYSTEM.SYSTEM.FASTP [4-69](#)

) SPOOLCOM prompt character [4-3](#)

_ PERUSE prompt character [3-6](#)