CVSI

CVSI, Inc.
4G Crosby Drive
Bedford, MA 01730

Operator's Guide to The Spooler Subsystem

Revision 23.0

DOC9303-4LA

Operator's Guide to the Spooler Subsystem

Fourth Edition

UK Technical Publications Team

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 23. 0 (Rev. 23)

Prime Computer, Inc., Prime Park, Natick, MA 01760

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer, Inc. Prime Computer, Inc., assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

PRIME, PR1ME, PRIMOS, and the PRIME logo are registered trademarks of Prime Computer, Inc. Prime INFORMATION CONNECTION, DISCOVER, INFO/BASIC, MIDAS, MIDASPLUS, PERFORM, PERFORMER, PRIFORMA, Prime INFORMATION, INFORM, PRISAM, PRIMAN, PRIMIX, PRIMEWAY, SIMPLE, Prime INFORMATION/pc, PRIME EXL, PRIME/SNA, PRIME TIMER, PRIMELINK, PRIMENET, PRIMEWORD, PRODUCER, PST 100, PT25, PT45, PT65, PT200, PT250, PW153, PW200, PW250, RINGNET, 50 series, 400, 750, 850, 2250, 2350, 2450, 2455, 2550, 2655, 2755, 6350, 6550, 9650, 9655, 9750, 9755, 9950, 9955, and 9955II are trademarks of Prime Computer, Inc.

Vendor Trademarks

DATAPAC is a registered trademark of Bell Canada.

DYNAPAC is a trademark of Dynapac Systems, 5350 Shawnee Road, Alexandria, VA 22312.

DATAPRODUCTS is a trademark of Dataproducts Corporation, 6200 Canoga Avenue, Woodland Hills, CA 91365.

DIABLO is a trademark of Diablo Systems, Incorporated, 901 Page Avenue, Fremont, CA 94537.

ETHERNET is a trademark of Xerox Corp, 800 Long Ridge Road, Stamford, CT 06904.

GANDALF Pin is a trademark of Gandalf Data, Inc, 1019 S. Noel, Wheeling, IL 60090.

GOULD is a trademark of Gould, Inc, 3631 Perkins Avenue, Cleveland, OH 44114.

QUME is a trademark of ITT Qume Corporation, 2350 Qume Drive, San Jose, CA 95131.

NEC is a trademark of NEC Information Systems, 1414 Massachusetts Avenue, Boxborough, MA 01719.

PostScript is a trademark of Adobe Systems.

PRINTRONIX is a trademark of Printronix, P.O. Box 19559, Irvine, CA 92713.

SCRIBE is a registered trademark of UniLogic Ltd.

TELENET is a registered trademark of GTE Telenet Communications Corp.

VERSATEC is a trademark of Versatec, Inc, 2710 Walsh Avenue, Santa Clara, CA 95051.

This document was prepared in the United Kingdom by Technical Publications Department,
4 Bramley Road, Mount Farm, Milton Keynes, MK1 1PT, United Kingdom.

PRINTING HISTORY

First Edition (DOC9303-1LA) February 1985 for Rev. 19.4
Second Edition (DOC9303-2LA) July 1987 for Rev. 21.0
Third Edition (DOC9303-3LA) October 1988 for Rev. 22.0
Fourth Edition (DOC9303-4LA) March 1990 for Rev. 23.0

CREDITS

Design: Richard Merry-West
Editorial: John Wells and Fiona Carey
Project Support: Vaughn Meads
Illustration: Will Mallender
Production: Prime Technical Publications unit.

CONTENTS

ABOUT THIS BOOK vii
How To Use This Book viii
Changes to the Spooler at Revision 23.0 ix
Related Documentation ix
Prime Documentation Conventions xi

1 INTRODUCING THE SPOOLER SUBSYSTEM 1–1
Introduction 1–1
How Files are Printed 1–1
Example Process 1–2
Enabling Users to Print Files 1–5
Routine Operational Tasks 1–6

2 SETTING UP THE SPOOLER SUBSYSTEM 2–1
Introduction 2–1
Preparatory Tasks 2–2
Directories and Files Used by the Spooler Subsystem 2–3
Printer Environments 2–7
Environment Attributes 2–10
Creating Character Mapping Files 2–16
Setting Up and Controlling Spool Queues 2–19
Subsystem Security 2–24
Compatibility With pre–Rev. 23.0 Spoolers 2–25
Print Handlers 2–27
Control of Laser Printers by PostScript Procedures 2–28
Cold Start Procedure 2–29
Setting Up the Electronic Vertical Format Unit (EVFU) 2–31
The Accounting Interface 2–37
Printout Distribution Addresses 2–40
Banner Page 2–41
Multi–language Despoolers 2–43

3 DEFINING PRINTER ENVIRONMENTS 3–1
Introduction 3–1
What an Environment File Is 3–1

How Environment Files are Used 3–2
Creating Environment Files 3–3
Summary of Printer Environment Commands 3–4

4 ROUTINE OPERATIONAL PROCEDURES 4–1
Introduction 4–1
Monitoring the Despooler Status 4–1
Controlling a Despooler Phantom 4–3
Changing to Another Printer Environment 4–5
Modifying a Printer Environment 4–6
Listing the Entries in the Spool Queue 4–7
Canceling the Printing of a File 4–7
Modifying Spool Requests 4–9
Hastening the Printing of a File 4–9

5 SOLVING OPERATIONAL PROBLEMS 5–1
Introduction 5–1
Dealing With Files That Will Not Print 5–1
Dealing With Mechanical Problems 5–2

APPENDICES

A SUMMARY OF SPOOL AND PROP COMMANDS A–1
Spool Command A–1
Prop Command A–12

B HOW TO WRITE A PRINT HANDLER B–1
Print Handlers B–1
Organizing the Work B–2
How the Despooler Software is Organized B–3
Functions Provided by a Print Handler B–4
Low–Level Device Drivers B–6
Details of Interfaces B–6
How to Build a Print Handler B–9
Data Structures B–10

C SPOOLER SUBSYSTEM MESSAGES C–1
Spooler Subsystem Messages C–2

D THE PDN SUPPORT OPTION D–1
Introduction D–1

E DESPOOLING OVER TCP/IP E–1
Introduction E–1

INDEX Index–1

ABOUT THIS BOOK

The Operator's Guide to the Spooler Subsystem is one of a set of Operator's Guides. Although it is called an Operator's Guide it is also for System Administrators or other people with responsibility for administering a computer system.

The Spooler subsystem controls the operation of your system's printers. This Guide describes and explains

- The structure of the Spooler subsystem
- How to set up the Spooler subsystem to meet your particular requirements
- How to deal with the day to day running of the Spooler subsystem.
- How to use the Spooler subsystem

You should be familiar with Prime systems before you read this book and in particular you should be familiar with the PRIMOS Operating system. If you have not used Prime systems before, you should read the *PRIMOS User's Guide*, which explains Prime's file management system and provides introductory and tutorial information about essential commands and utilities. It also explains how the SPOOL command can be used.

Before studying the Spooler subsystem you may find it useful to read the *Operator's System Overview*, particularly if you are not familiar with the terminology that is used in Prime systems.

How To Use This Book

This book contains five chapters and five appendices.

Chapter 1

Introducing the Spooler Subsystem contains a general introduction to the Spooler subsystem. It explains what the subsystem is for, shows how it is used, and outlines the way that the software operates. It lists the tasks you perform to set up the Spooler subsystem, and the tasks that are necessary in its day to day operation.

Chapter 2

Setting Up the Spooler Subsystem, explains the components of the Spooler subsystem in greater detail. It gives technical information that enables you to set up the Spooler subsystem to provide the type of printing services that you need. The information is at a suitable level for a System Administrator, systems analyst, or programmer.

Chapter 3

Defining Printer Environments, gives more detailed instructions on how to set up printer environments. The technical level is similar to that of Chapter 2.

Chapter 4

Routine Operational Procedures, gives procedures for commonly performed operations that use the SPOOL and PROP commands. The information is appropriate for operators and other users and is referenced by the operation that you want to perform, rather than by the command that you use.

Chapter 5

Solving Operational Problems, explains how to deal with possible hardware and software problems that can arise.

Appendix A

Summary of Spool and Prop Commands, gives a summary of the SPOOL and PROP commands. All available options are listed for reference.

Appendix B

How to Write a Print Handler, explains how to add support for non-standard output devices.

Appendix C

Spooler Subsystem Messages, lists all the messages that can be sent out by the Spooler subsystem , and gives extra information for some of the messages, where necessary.

Appendix D

The PDN Support Option, describes how you can use the Spooler subsystem to send files to printers that are attached to a Public Data Network (PDN). The software that enables you to print files over a PDN is available as a separately priced option.

Appendix E

The TCP/IP Support Option, describes how you can use the Transmission Control Protocol/Internet Protocol support software to use despooler to control printers connected across the Prime TCP/IP network.

Changes to the Spooler at Revision 23.0

The following new functionality is included in the Spooler subsystem at PRIMOS Revision 23.0.

At Rev. 23.0 the module for displaying the spool queue entries can be replaced by your own routine. If you do not supply your own routine then the existing routine called by SPOOL -LIST will continue to be used as normal.

The display from SPOOL -LIST -BRIEF operations is configurable, based on ASCII files defining the items to be displayed.

Related Documentation

Other Prime documentation that you may find useful includes the following:

- *System Administrator's Guide, Volumes I, II,* and *III* (DOC10131-2LA, DOC10132-2LA, DOC10133-2LA) which describe how to set up, configure, and maintain PRIMOS.
- *Operator's System Overview* (DOC9298-3LA), which introduces the series and describes computer-room operation of Prime systems.
- *Operator's Guide to System Monitoring* (DOC9299-3LA), which describes how to monitor system activity, messages, and utilization, and explains the messages that are sent to the supervisor terminal. The System Administrator needs this manual to alleviate problems of poor system response time.
- *Operator's Guide to File System Maintenance* (DOC9300-5LA), which describes: the PRIMOS file system; how to format disk partitions; how to run the disk partition maintenance program; how to determine physical device numbers; and how to interpret disk error messages.
- *Data Backup and Recovery Guide* (DOC10129-2LA), which describes how to save information on disk or tape, and how to restore that information later.
- *Operator's Guide to System Commands* (DOC9304-5LA), is a reference guide that lists many of the operator commands and their options in alphabetical order. It documents most of the commands that are described in the other books in this series.
- *Operator's Guide to Batch* (DOC9302-3LA), is a guide to both the operation and administration of your batch subsystem.
- *Programmer's Guide to BIND and EPFs* (DOC8691-1PA), which describes the BIND linker and Executable Program Formats (EPFs).

Before studying the Spooler subsystem you may find it useful to read the Operator's System Overview, particularly if you are not familiar with the terminology that is used in Prime systems.

Other Relevant Manuals

Other manuals that you may find useful are

- The documentation for your CPU, and the "Using Your" series of books.
- The documentation for letter quality printers, *LQP 3185 Guide* (DOC7634-001L) and *LQP 3175 Guide* (IDR5024).
- *PRIMOS Commands Reference Guide* (DOC3108-7LA), and (UPD3108-6LA) which provide detailed information for user commands.

Prime Documentation Conventions

The following conventions are used in command formats, statement formats, and in examples throughout this document. Command and statement formats show the syntax of commands, program language statements, and callable routines. Examples illustrate the uses of these commands, statements, and routines in typical applications. Terminal input may be entered in either upper case or lower case.

Convention	*Explanation*	*Example*
UPPERCASE	In command formats, words in uppercase indicate the actual names of commands, statements, and key words. They can be entered in either uppercase or lowercase.	PROP -VERIFY
lowercase	In command formats, words in lowercase represent items for which you must substitute a value.	SPOOL myfile
Abbreviations	The abbreviation for a command or statement is underscored. If the command or directive itself contains an underscore, the abbreviation is shown below the full name.	ATTRIBUTE -FAST_INTERNAL -FI
Underscoring in examples	In examples, user input is underscored but system prompts and output are not.	OK, PROP -STATUS
Brackets [ ]	Brackets enclose a list of one or more optional items. Choose none, one or more of these.	SPOOL xx [-NOFTM -SFI]
Braces { }	Braces enclose a vertical list of items. Choose only one of these.	PROP {-HELP -STATUS}
Ellipsis ...	An ellipsis indicates that the preceding item may be repeated.	item-x [, item-y] ...
Hyphen -	Wherever a hyphen appears in a command line option it is a required part of that option.	SPOOL -LIST
Angle Brackets < >	Angle brackets are used to separate the elements of a pathname.	<MFD>SPOOL*>DOC
Octal ' or ^	The ' or ^ symbol shows that the number is given as an octal value.	'240 or ^240 are equvalent to 160 (base 10)

1

INTRODUCING THE SPOOLER SUBSYSTEM

Introduction

This chapter gives a brief overview of the Spooler subsystem. The chapter summarizes the administrative tasks that must be performed before the spooler can be used, and also describes the routine operational tasks. It shows how the Spooler subsystem appears to a user and states, in general terms, the steps to follow to print a file. The chapter also describes the types of file that can be printed, and introduces the concept of printer environments. An example of a typical print request is then followed through the system.

How Files are Printed

The Spooler subsystem allows users of a Prime system or network to print their files. It enables them to use printers in an efficient and organized manner.

To print a file the user must issue a SPOOL command. The SPOOL command has many options that enable the user to give print directives such as:

- Defer printing to some later time.
- Specify the site at which the job is to print.
- Ask for a number of copies to be printed.

When a SPOOL command is issued the Spooler subsystem places the print request in a queue of requests that wait for attention. The file is printed as soon as a suitable printer is available.

The User's Point of View

It is not necessary to know anything about how the Spooler subsystem works to use the SPOOL command and to have files printed. Most users need only know a few of the many options of the SPOOL command so that they can specify the type of printer, paper, and format that they want.

To obtain a printout from the computer system the user must perform the following four distinct steps:

1. Obtain or create a printable file.
2. Issue the correct SPOOL command.
3. Check on the progress of the request.
4. Collect the printed output.

The File

Any file can be printed irrespective of its origin. A file could be a text file created with a screen editor such as EMACS or with a text formatter such as RUNOFF. Such a file could contain a business letter, source code for a computer program, a technical document or similar text. Alternatively a file could contain company information, electronic mail, or the output of an application program. You can spool special files for plotters or laser printers as well as ASCII text files. However, it is essential to specify the correct printer environment by using one of the SPOOL command options.

The Printer Environment

There are usually several printers available on a computer system, and at different times they may be using different types of paper, or producing a different page format. The combination of printer type, location, paper type, format and many other factors is called a **printer environment.**

To select a particular environment, named **attributes** of the environment are included in the SPOOL command. The System Administrator should tell users what attributes can be specified on the system. It is most important to grasp the idea that the attributes in a SPOOL command cause the request to be handled by a suitable environment, and not necessarily by a specific printer. It may well be that in a small establishment, with few printers, specifying one or two attributes is equivalent to specifying the printer to be used, but in a larger system there could be several environments that meet your needs, and the file could be printed on any one of a number of printers.

Often, users need not be concerned about the environment that deals with a request. If the System Administrator has set up your system to provide default attributes it is sufficient to give a SPOOL command with just the path name of a file that is to be printed. The Spooler subsystem then provides default attributes that select the standard printer and paper type on your system.

Example Process

This section first describes the process of spooling a file from the user's point of view. It then describes the same example from a software point of view.

Suppose that you prepare a business letter with RUNOFF, and file it in your current directory with the filename LETTER.RUNO. The letter now needs to be printed on a letter quality printer.

Assume for this example that several environments are available and that one of those serves a QUME daisy wheel printer. The System Administrator would set up the Spooler subsystem such that an attribute of QUME in a SPOOL command selects the QUME printer. For an explanation of the command line syntax see Appendix A, Summary Of Spool And Prop Commands.

Submit Request

The required SPOOL command line would be as follows:

```
SPOOL  LETTER.RUNO  -ATTRIBUTE QUME
```

The -ATTRIBUTE option can also be entered as -ATT. Many other options could be given in the command line, for example, -DEFER to defer printing to a later time, or -COPIES to request several copies. The available options are listed in Appendix A, Summary of Spool and Prop Commands. When you issue the SPOOL command, the software checks to see whether the file can be found and whether all the specified options are acceptable. If the System Administrator has created one or more attributes files, then all the attributes specified in the command are checked to make sure that they are legal.

If the file exists, and the options are specified correctly, the software puts details of the print request into the **spool queue** (a list of print requests awaiting attention), notifies the user that the file is in the queue, and tells the user the identifying number of the request.

For example:

```
OK, SPOOL LETTER.RUNO -ATTRIBUTE QUME
[SPOOL Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Request 4 added to queue, 1 records : <SPOOLA>FRED>LETTER.RUNO
```

Possible Errors

If the request is not accepted because of a mistake in the command line it should be re-typed correctly. If an illegal attribute has been used the user must either use a different combination of attributes or contact the System Administrator to obtain the right to use certain attributes. The SPOOL command option -SPOOL_WHILE_OPEN can only be used if the System Administrator has enabled its use for the user.

Check Progress of Job

This step is optional. While the request is in the queue, or while it is printing, its progress can be viewed with the SPOOL -LIST command, as illustrated in the following example:

```
OK, SPOOL -LIST
[SPOOL Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
System SYSA
Request  Time  User      File            No Size  State
-------- ----  --------- --------------- -- ----- -----
4        2104  FRED      LETTER.RUNO     1  1     Print
5        2106  OPS.7     TUE-07-08       1  36
6        2107  OPS.7     WED-07-09       1  38
7        2108  OPS.7     THU-07-10       1  35
```

Retrieve the Printed File

When this file is printed, the user retrieves the printed file, either from the printer itself or by following the procedures established by the System Administrator.

What the Software is Doing

When a SPOOL command is given the software checks it carefully. If there are attributes in the command they are checked against special lists of attributes the System Administrator has created. There may be a general list for the whole system, or special lists for individual users or projects. The software also checks that the file specified in the SPOOL command exists.

The request is then put into a queue in the SPOOL_QUEUE* directory. The file is copied into one of the SPOOL_DATA* directories, unless the -NO_COPY option was used. See Chapter 2, The SPOOL_DATA* Directory, for further details.

The request and file then wait until a **despooler phantom** processes the request. A despooler phantom is a process that controls one of the printers (for general information on phantoms see the Prime User's Guide). Several despooler phantoms can run concurrently. Each phantom searches in the spool queue, at predefined intervals, for jobs that it can handle. Each despooler can search for jobs in queues on other systems in the same network, as well as in the local queue. The despooler phantom recognizes requests that it can process by examining their attributes.

When a request is put into the queue, the SPOOL command notifies the appropriate despooler phantom, so if the despooler is not checking a queue or printing a job, the file is printed immediately. However, if the despooler is printing another file, the job waits its turn. The process is summarized in Figure 1-1.

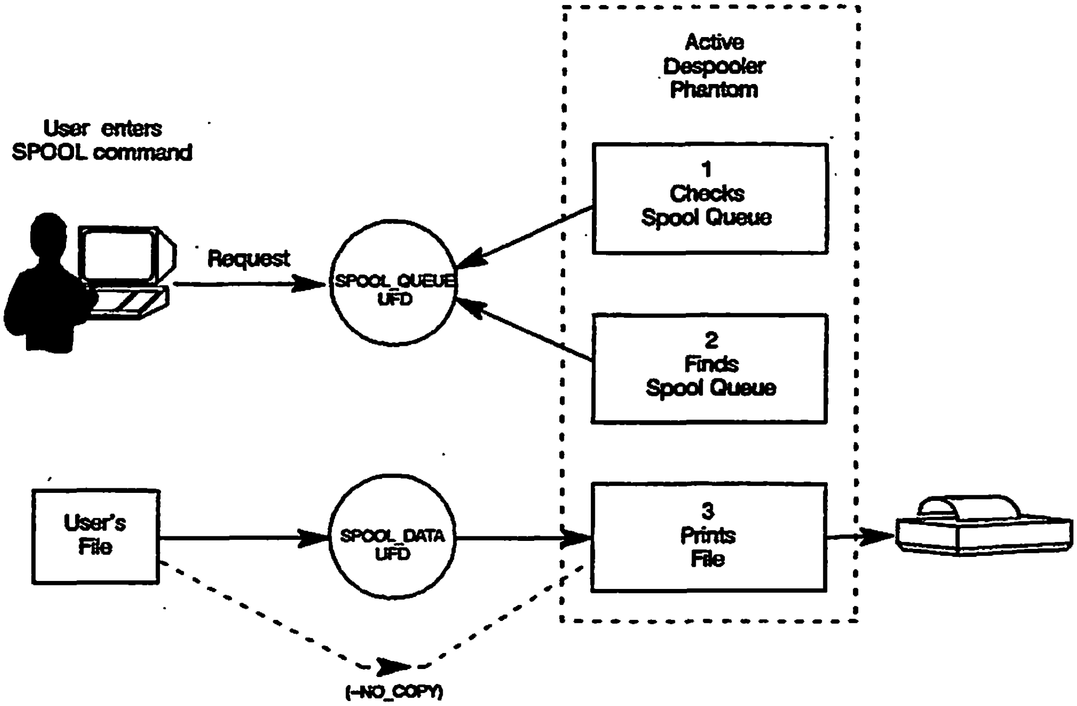

FIGURE 1-1 Spooling a File

Enabling Users to Print Files

The System Administrator, or another person responsible for the system, must perform a number of tasks before the Spooler subsystem can be used. These tasks are listed below, and explained fully in Chapter 2, Setting up the Spooler Subsystem.

The tasks are as follows:

- Create a number of special files.
- Create Access Control Lists (ACLs) for the files and directories that the Spooler subsystem uses.
- Define the printer environments that are needed to meet users' printing needs.
- Specify privileged users by making them members of the .SPOOL_ADMINISTRATOR$ ACL group, thus appropriately including them in the SPOOL*>RESTRICTED_USERS>SWO_VALID_USERS file
- Ensure that the system cold-start procedure starts up the necessary despooler phantoms.

As well as these purely technical aspects of setting up the Spooler subsystem, the System Administrator has to allocate daily work routines for operators, and standard procedures for the users. In particular the System Administrator must

- Set up the schedules that decide when each environment will be available, and when and how printouts will be distributed.

- Make sure that operators and users know of these schedules and environments, and that they have any other knowledge they need in order to use the system successfully.

The System Administrator should have a clear idea of what procedures and schedules are needed before setting up the system. Administrative decisions affect what environments it is sensible to define. For example, the way that listings are distributed influences the choice of default attributes, and the information that appears in the header pages of listings; methods of accounting could affect what information is put into trailing pages of listings; and the prices of different types of printer paper could influence the choice of default attributes.

Routine Operational Tasks

Some operational tasks should be part of a regular routine, whereas others must be performed as required. Operators frequently need to perform the following tasks:

- Stop, start, and restart environments.
- Change printer environments to print special requests that need unusual formats. (To see if there are any such requests use either the SPOOL -LIST -ALL -DETAIL command or set up a configurable SPOOL -LIST display). It is good procedure to schedule the print of special forms for specific times of day.
- Reload or change paper and ribbons in the printers as necessary, for example, when changing an environment.
- Remove listings from the printer, and distribute them.
- Monitor the spool queue with SPOOL -LIST -ALL to check on the status of print requests on local and remote systems.
- Help users to modify print requests as necessary.
- Drop print jobs that were spooled by mistake and have already begun to print.
- Maintain printers according to the manufacturer's instructions. For example, vacuuming a line printer at least twice a day, and similar tasks.

For more information on operational tasks, see Chapter 4, Routine Operational Procedures, and Chapter 5, Solving Operational Problems.

2

SETTING UP THE SPOOLER SUBSYSTEM

Introduction

This chapter is intended for the System Administrator or other person with responsibility for setting up and administering the Spooler subsystem. Before you read this chapter, you should read Chapter 1, Introducing the Spooler Subsystem, for an overview of the subsystem.

This chapter explains the following features:

- The file structure of the Spooler subsystem
- Printer environments and environment attributes
- How spool queues are managed
- Subsystem security
- Compatibility with pre-Rev. 21.0 spoolers
- Print handlers
- Despooler phantoms and how to use them
- The Electronic Vertical Forms Unit (EVFU)
- The accounting routine
- Banner pages, and distribution addresses
- User configurable SPOOL -LIST displays

Preparatory Tasks

To make the Spooler subsystem ready for use you need to perform the following tasks:

1. Read the INFO FILE for an explanation of the installation procedure.
2. Create suitable environments for printers and plotters. You can use a text editor such as ED or EMACS, the PRIMOS file handling commands, and the PROP -VERIFY command to create, modify, verify, or delete environment files. Environment files contain information that is needed by the despooler phantoms to enable them to search spool queues for suitable jobs, and to ensure that files are printed correctly. A utility program is provided that automatically produces templates of Rev. 21.0 environment files from pre-Rev 21.0 environments.
3. Create attributes files so that attributes specified in SPOOL commands can be checked, and so that default attributes can be supplied if necessary. Although attributes files are useful, they are optional, and the subsystem can operate without them.
4. Create the SPOOL_DATA* directories that you need, and create the DATA_PARTITIONS file in SPOOL_QUEUE*. Set the correct ACLs on the subsystem files and directories. A utility program is provided that sets the correct ACLs for you. You should use this utility unless you need different ACLs for reasons of your own.
5. Review whether any users need the ability to see all the entries in the spool queue. This aspect of the Spooler subsystem is controlled by the FULL_LIST_USERS file in the SPOOL_QUEUE* directory.
6. Create privileged users by adding their userIDs to the .SPOOL ADMINISTRATOR$ ACL group, and to the SPOOL*>RESTRICTED_USERS>SWO_VALID_USERS file, if appropriate. Privileged users can control the Spooler and use all the options of the PROP and SPOOL commands subsystem.
7. Create Electronic Vertical Format Unit (EVFU) files for printers that need them.
8. Set up a suitable cold-start procedure. To do this you must create a section of the PRIMOS.COMI file that automatically starts the appropriate despooler phantoms when a cold start of the system is performed.
9. Create a file of distribution addresses that correspond to userIDs. This is optional.
10. Set up configurable SPOOL -LIST displays as required. This is optional.

Information for Users

The flexibility of the Spooler subsystem enables you to meet the precise needs of your organization. However, this very flexibility means that it is not possible for a manufacturer's manual to give users all the information that they need to use the SPOOL command, since much depends on how you choose to use the features that are available. You will need to make sure that users, especially non-technical users, know about the following:

- The attributes that they can specify, and how to use them
- Restrictions on viewing the spool queues
- The particular SPOOL command options that they are likely to need

Directories and Files Used by the Spooler Subsystem

Runfiles for the SPOOL and PROP commands are on the command device in CMDNC0. In addition, the Spooler subsystem uses three special top-level directories: SPOOL_QUEUE*, SPOOL_DATA*, and SPOOL*. These directories must be protected by ACLs. You can have several SPOOL_DATA* directories if necessary, but you must create them yourself. For details of how to set up several SPOOL_DATA* directories see the later section of this chapter, Setting Up and Controlling Spool Queues.

The contents of the directories used by the Spooler subsystem, and recommended ACLs for them, are summarized below, and in Figure 2-1. For information on the .SPOOL$$ and the .SPOOL_ADMINISTRATOR$ ACL groups, see the later section of this chapter, Subsystem Security.

The SPOOL_QUEUE* Directory

The SPOOL_QUEUE* directory contains a file called QUEUE. This file holds the list of print requests that are awaiting attention. The SPOOL_QUEUE* directory can also hold two optional files:

FULL_LIST_USERS file — Controls who can see the whole spool queue with the SPOOL -LIST command.

DATA_PARTITIONS file — Contains a list of names of disk partitions on the local machine on which SPOOL_DATA* directories have been created. For full details see the later section of this chapter, Setting Up and Controlling Spool Queues.

The ACL for the SPOOL_QUEUE* directory must be as follows:

```
.SPOOL$$:ALL
$REST:NONE
```

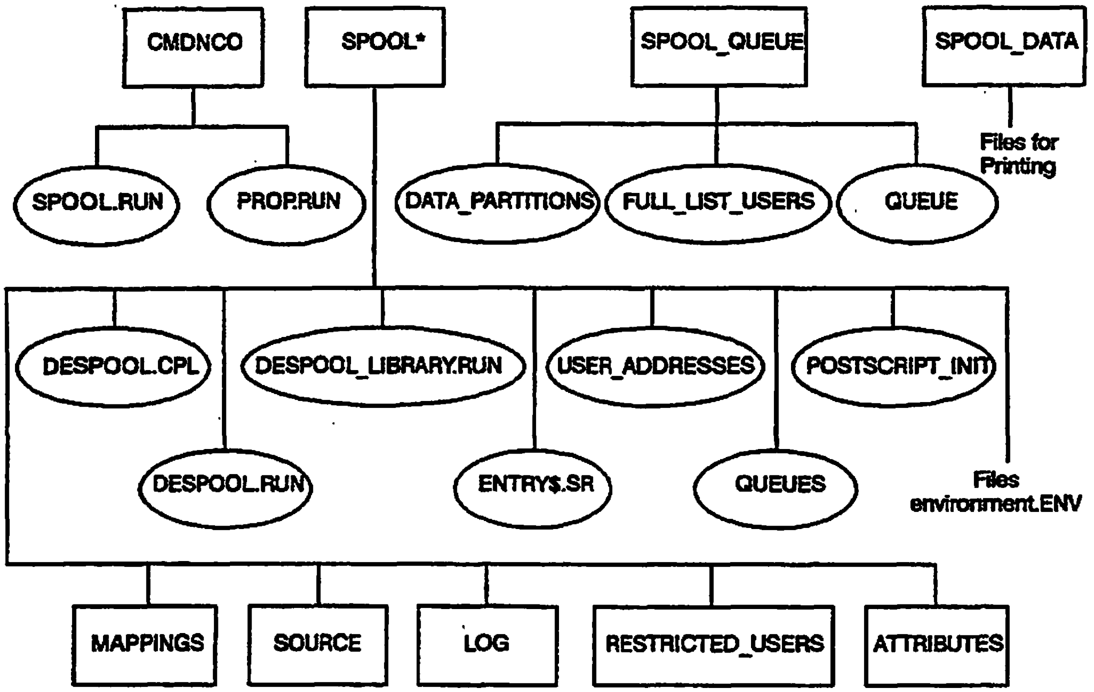

FIGURE 2-1 Directories and Files Used by the Spooler Subsystem

The SPOOL_DATA* Directory

The SPOOL_DATA* directory holds copies of the files that are waiting to be printed. However, files are not copied into SPOOL_DATA* if the SPOOL command uses the -NO_COPY or -SPOOL_WHILE_OPEN options. -NO_COPY is for very large files that would occupy excessive disk space in SPOOL_DATA*. -SPOOL_WHILE_OPEN allows specified users to spool a file, and start to print while the file is still being written to disk.

The ACL for the SPOOL_DATA* directory must be as follows:

```
.SPOOL$$:ALL
$REST:NONE
```

The SPOOL* Directory

The SPOOL* directory holds all other spooler subsystem files and sub-directories. You must protect this directory with an ACL that gives LUR access to the average user and ALL access to members of the .SPOOL_ADMINISTRATOR$ group and System Administrators.

The contents of the SPOOL* directory are as follows:

Pathname — *Description*

SPOOL*>DESPOOL.CPL
A CPL file, used by the PROP command software, that controls invocation of the despooler program. You can modify this file to add ADD_REMOTE_ID (ARID) commands for network access. You should prevent users from seeing the ARID passwords by setting an ACL that permits only the .SPOOL_ADMINISTRATOR$ group to access this file.

SPOOL*>DESPOOL.RUN
The runfile for despooler phantoms. This runfile is invoked each time a despooler phantom is started.

SPOOL*>DESPOOL_LIBRARY.RUN
The library of print handler routines supplied by Prime.

SPOOL*>ENTRY$.SR
Sets the search rules for a despooler phantom. The file as supplied by Prime contains the following lines:

```
SPOOL*>DESPOOL_LIBRARY.RUN
-SYSTEM
```

This file is essential and you should modify it only as instructed in Appendix B and D of this guide. If you supply your own accounting routine, banner page routine, or print handlers, you must add the location of the EPF entrypoints to the search rules in this file.

SPOOL*>MAPPINGS
A subdirectory that holds the character mapping files ECS_TO_ASCII_MAP and QUME_OAS_MAP. The first file maps ECS to ASCII, while the second is a map for QUME printers, that require OAS support and ECS to ASCII. This subdirectory can also hold the STANDARD_MAP file, although the file is not placed there by the PRIMOS default build.

SPOOL*>POSTSCRIPT_INIT
The default file used when a PostScript printer is initialized. You should list in this file the pathnames of files that contain PostScript procedures.

SPOOL*>QUEUES
An optional file. It is structured similarly to an environment file, as described later in the chapter and can use the following commands:

```
DISK
DISK_SCAN
NODE
POLL_RATE
PRIORITY_TO_LOCAL_QUEUE
```

The file has two uses:

First, the SPOOL -LIST -ALL command reports on all the queues defined in this file. If there is no SPOOL* > QUEUES file then SPOOL -LIST -ALL reports on the local queue only.

Second, despooler phantoms can refer to the information in SPOOL* > QUEUES by including the command

```
CONFIG QUEUES
```

in each environment file. If you specify queue information by this method you can avoid the need to duplicate information in each environment file, and thus can make spool queue management easier.

SPOOL* > RESTRICTED_USERS

A sub-directory that holds the file SWO_VALID_USERS which contains a list of authorized spool-while-open users. These users are identified one per line by login IDs, ACL groups, or the .ALL_USERS. category.

SPOOL* > ATTRIBUTES

A sub-directory that holds attributes files. Attributes define characteristics of printers, or printer environments that users can specify in their SPOOL commands. Attribute files hold lists of valid attributes. They are described in detail later in this chapter.

SPOOL* > SOURCE

Source code and insert files for use in your own routines.

SPOOL* > LOG

A sub-directory that holds log files for the despooler phantoms. For full details see the description of the LOG command in Chapter 3, Defining Printer Environments.

SPOOL* > USER_ADDRESSES

An optional file that contains distribution addresses for different user IDs. For full details see the later section of this chapter Printout Distribution Addresses.

SPOOL* > CONFIG_LIST

A subdirectory for holding default SPOOL -LIST configuration files as well as such files for each user and project.

Environment files are also held in SPOOL*. All environment files have pathnames of the following form:

```
SPOOL* > environment.ENV
```

where environment is a name you have chosen. Note that you must provide the .ENV subscript to the filename. Printer environments are explained in the next section of this chapter. For details of environment file commands see Chapter 3, Defining Printer Environments.

When each despooler phantom is first started, a file with the pathname SPOOL* > environment.SEM is created for it. These files enable the SPOOL command, after adding a request to the spool queue, to activate the correct despooler phantom. Files with the .SEM component must not be altered in any way. The .SEM stands for semaphore, and refers to the way that inactive processes are queued by the operating system.

Note that any EVFU files you create must also be stored in SPOOL*.

Printer Environments

It is important to distinguish clearly between the terms printer, printer environment, environment file, and despooler phantom.

The **printer** is the hardware component that does the printing.

The **printer environment** defines all aspects of how a printer is used, and of how print requests made by SPOOL commands are matched with suitable printers. Some features of the printer environment are clearly visible to the user, but others are not.

Features of the environment that are visible to the user include:

- The type of paper that must be present
- The types of file the environment can handle
- The range of file sizes the environment is permitted to print
- How the output is formatted
- Whether or not to convert text to upper case

The features that are not apparent to the average user include the locations of the print queues that are to be searched, how frequently to check the queues for work, and how the printer is connected to the computer.

An **environment file** contains all the information that is needed to implement a printer environment. It includes technical details of the output device, commands that control the header and trailer pages of printouts, information about spool queues and so forth. Environment files are stored in the SPOOL* directory. The SPOOL* directory is protected by an Access Control List (ACL). Consequently the environment files are automatically secure. For details of environment files see Chapter 3, Defining Printer Environments.

A **despooler phantom** is a process that controls a printer environment, and consequently controls the printer that is specified in the environment.

How to Use Printer Environments

It is important to set up printer environments that are suitable for the kinds of printing that users of the system need.

Each printer can have a number of printer environments, but can only run in one environment at a time. You may have several environments on your system, even if you have only one printer.

There are many reasons why you might want to set up extra environments. Some possible reasons are as follows:

- Different types of paper. For example, a fast line printer could have a number of environments filed in SPOOL*:
  - Environment PR0 using standard wide listing paper and a standard print format.
  - Environment WHITE using narrow white paper; environment LAND, using landscape-oriented paper; and environment SHIP, to print shipping labels on special forms.
  - Whenever the type of paper in a printer is changed a different environment should be started.
- Different types of fonts. You can create separate environments to change the font, or to map the characters you see at your terminal to be different at the printer.
- Various sizes of file. During working hours, you may want to limit the size of files that are printed. In the evening you may wish to impose no limits at all. To put the restrictions into effect, you would create two environments, one for each situation, and use the appropriate one at each time.
- Different printer speeds. If your installation has two printers of differing speeds, you may want to send all large files to the faster printer. To do this you would limit the size of files that the slower printer's environment would accept.
- Default settings. With this wide choice of environments you should decide on the default settings for paper orientation, paper bin, font type, and so on, that are best suited to your site requirements.

You might decide that some environments, those for instance that deal with special formats or require special paper, are used so rarely that they should only be started up by special request. If you are short of printer resources a convenient solution might be to organize a rota of commonly used environments.

Note

> When you have defined the environments that are needed on the system you should not change them. If you need to use a printer in a slightly different way you should not alter the commands in an environment file and restart the environment. This could introduce errors in tried and tested environment files, and besides, is inconvenient. Instead, when a printer needs to be used differently you should stop the active environment and start an alternative one. In this way an operator, who need have no knowledge of how environment files are structured, can easily, and as a routine matter, change the active environment for a printer.

Conversion of Pre-Rev. 21.0 Environment Files

A utility is provided that produces a Rev. 21.0 environment file template from an existing pre-Rev. 21.0 environment file. At Rev. 21.0 you may need several slightly different versions of each environment. In that case, use the utility described below to produce a template that you can modify to create each of the environments you need.

The command to invoke the utility is as follows:

```
R SPOOL* > CONVERT_ENV  [    old_env new_env             ]
                        [  { -ENVIRONMENT }  name        ]
                        [  { -ENV         }              ]
                        [    -HELP                       ]
```

where:

old_env is the pathname of a pre-Rev. 21.0 environment file.

new_env is the pathname of the file in which the Rev. 21.0 environment file is to be stored.

name is an environment file name without either the E. prefix or the .ENV suffix.

If you use the -ENVIRONMENT option the utility looks for the file SPOOLQ>E.*name* and produces the file SPOOL*>*name*.ENV from it. If you specify pathnames, you have complete freedom to file both the old and new files wherever you wish in the file system.

Environment File Commands

The features of a printer environment are specified by corresponding commands in the environment file. The commands are considered in five groups as described below. You must consider each group when you plan your environments. For full details of environment file commands see Chapter 3, Defining Printer Environments.

Command Type	*Description*
Scheduling commands	Commands in this group match each Spool request with an appropriate environment, and determine the order in which files are printed. For details of how environments are matched with Spool requests, see the section below entitled *ENVIRONMENT ATTRIBUTES.*
Spool queue commands	Commands in this group deal with such matters as spool queue location and how frequently each spool queue is scanned by the despooler phantom.
Formatting commands	Commands in this group specify how the printed output is formatted. They include reference to specific print handler routines. It is possible to add printer handler routines for new output devices.
Interface commands	Commands in this group give details of the type of output interface, data transmission protocol, and so on.
LOG command	Allows you to log environment details. You can choose to keep an historic log in which environment details are recorded whenever an environment is invoked, or to keep details of only the most recent invocation of an environment.

Viewing Printer Environments

You can view the contents of an environment file by giving the command

```
PROP environment -DISPLAY
```

Alternatively you can use a text editor such as ED or EMACS.

To see what environments are available on a system you can give the command

```
PROP -STATUS -ALL
```

This command also shows the status of the environment. Alternatively, you can attach to the SPOOL* directory and list its contents.

Environment information is also available in the SPOOL*>LOG sub-directory. If you have mistakenly altered an environment file you can obtain information about its previous contents from the log.

The example below shows a simple environment file. For a more detailed example, with explanatory comments, and for full details of the environment file commands, see Chapter 3, Defining Printer Environments.

```
PARALLEL PR0
DEVICE PRINTRONIX -EVFU_OFF
PRIORITY_TO_LOCAL_QUEUE
FORMAT -W78 -L64
MESSAGE Lined Wide Paper Printer
LOG -SAVE
ATTRIBUTE PLOT -MANDATORY
ATTRIBUTE SITE_A
NODE -RING
```

Environment Attributes

Important features or properties of output devices, such as the type of paper in a printer, or the location of a printer, are called attributes. Strictly speaking, attributes are features of the printer environment, but attributes that correspond directly to some feature of the output device can loosely be called printer or device attributes. The particular attributes of an environment are specified in the environment file with the ATTRIBUTE command.

In their print requests users specify attributes that the printer must have with the -ATTRIBUTE (-ATT) option of the SPOOL command. The spool request is then handled by an environment that has all the attributes named in the SPOOL command. You should set up environment files so they contain environment attributes which conform to the real features of the printer which is served by your system. You must also inform users of the names you have chosen of the printer attributes that they may specify.

environment's mandatory attributes. The print request need not include all or any of the environment's non-mandatory attributes. However if the print request contains attributes not in the environment file, the request will not be processed.

Example of the Use of Attributes

A simple example illustrates the use of attributes. Suppose that a system has two line printers, PR0 and PR1, one using wide paper, the other using document paper. Suppose also that the environment file for the first printer, PR0.ENV, states WIDE as an attribute, and the corresponding command in the environment file PR1.ENV states DOC as an attribute.

The command to print a file on wide paper would be as follows:

```
SPOOL filename -ATTRIBUTE WIDE
```

Similarly, the command to print a file on document paper would be as follows:

```
SPOOL filename -ATTRIBUTE DOC
```

Prior to Rev. 21.0 the SPOOL command options -FORM, -AT, and -TYPE were used to specify form type and printer location. At Rev. 21.0 and after, you can use the -ATTRIBUTE option to specify these features and many other types of attribute as well.

Attributes Files

Attributes files are an optional feature of the Spooler subsystem that you can use to

- List valid attribute names so that the attributes specified in SPOOL commands can be checked. If no attributes file exists, the attribute names specified in SPOOL commands are not checked.
- Create groups of mutually exclusive attribute names. The SPOOL command rejects any attempt to use two or more attribute names from the same group.
- Create default attributes, to be used if no attribute from a group is specified in a SPOOL request. You must create at least one attributes file if you want the Spooler subsystem to provide defaults automatically. If no attributes file exists then default attributes cannot be supplied.
- Make alternative printing facilities available to different users by creating different sets of valid attributes for different users or user groups.

Naming Attributes Files

To set up attributes files, you must first create the directory SPOOL*>ATTRIBUTES. In this directory you can create any number of attributes files. Permitted file names are as follows:

Type of List	*Attributes File Name*	*Example*
DEFAULT	.DEFAULT_ATTRIBUTES.	
User	User-ID	FRED
Project	*project_name	*PROJECT

How the Correct Attributes File is Found

When the attributes in a SPOOL command are checked the Spooler first looks for a user-specific attributes file. If no such file is found it looks instead for a file with the project name under which the user is logged in, but with an asterisk prefix. The asterisk enables the SPOOL command software to recognize which files contain attributes for project groups, and also makes it easy for you to distinguish the different types of attribute file. If neither a user-specific nor a project specific-attributes file is found then the SPOOL command looks for the default attributes file.

If the system has no attributes files, whatever text strings were specified as attributes are accepted without validation. Naturally if they do not correspond to any real environment attributes the print request remains in the queue until it is canceled or corrected.

Contents of Attributes Files

An attributes file is a list of attribute names. Groups of attribute names are separated by at least one blank line or a line containing a comment. The first name in each group is the default attribute that is used if no attribute from the group is given in the SPOOL command. If no default is required, the reserved name .NO_DEFAULT. must be given at the start of the group. Note that periods before and after .NO_DEFAULT. are part of the name.

You can include comments by prefacing them with the /* marker. A comment can occupy a line by itself, or may appear after an attribute name. If a comment is the only item on a line, the line then separates groups of attributes, just as a blank line would.

Example of Attributes Files

Consider a system with two sites, three computers, and five printers as shown in Figure 2-2.

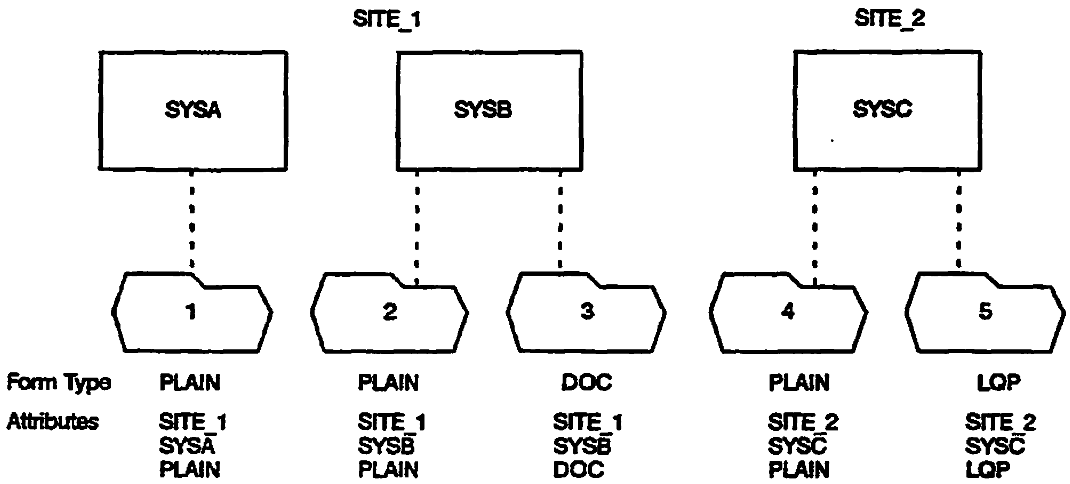

FIGURE 2-2 Example of Attributes in Systems in a Network

The attributes files for SYSA and SYSB would both be as follows:

```
/* Site names
SITE_1
SITE_2
/* Form types
PLAIN
DOC
LQP
/* System names
.NO_DEFAULT.
SYSA
SYSB
SYSC
```

On SYSC the default site should be SITE_2, therefore the attributes file for SYSC would be as follows:

```
/* Site names
SITE_2
SITE_1
/* Form types
PLAIN
DOC
LQP
```

```
/* System names
.NO_DEFAULT.
SYSA
SYSB
SYSC
```

Note that none of the attributes in the attributes file need to be specified as mandatory in the printer environment files. For each group of attributes, either a default is provided from the attributes file, or it does not matter that an attribute is not specified. For further information, see Chapter 3, Defining Printer Environments.

To see how this system of attributes works in practice, assume that there are five active despooler phantoms, one for each printer, and that each phantom searches all of the spool queues. The examples below show how different combinations of attributes select different printers.

Example 1:

A user on SYSA issues the command

```
SPOOL file-name
```

The default attributes are SITE_1 and PLAIN. Thus the document is printed by either printer 1 or printer 2, depending on which of the two suitable despooler phantoms happens to remove the request from the queues.

Example 2:

A user on SYSC issues the command

```
SPOOL file-name -ATTRIBUTE DOC
```

The default attribute SITE_2 is supplied. There is no despooler phantom that provides form type DOC at SITE_2, so the request remains in the queue indefinitely. It is up to the System Administrator to make sure that his users know what attributes to use, or to set up the system so that they cannot make incorrect commands like this. The user should have given the command

```
SPOOL file-name -ATTRIBUTE DOC SITE_1
```

This causes the document to be printed by printer 3.

Example 3:

A user on SYSB issues the command

```
SPOOL file-name -ATTRIBUTE SITE_2 PLAIN
```

The file is printed by printer 4.

Each of the attributes used in this hypothetical system describes a real physical attribute of the printer; where it is located, which computer it is attached to, and what type of paper it uses.

For convenience, attributes can be included in the environment that have no corresponding feature in the printer itself. Suppose that you have a single computer, and a single printer that serves a five-floor building. To simplify the distribution of printed output you would wish to print files for one floor at a time.

To implement this plan you would need to create five environment files for the printer. These files would be identical except for the attribute that states the floor number. You would also need to add the following lines to the attributes files on your system:

```
 /* Floor numbers
.NO_DEFAULT.
FLOOR_1
FLOOR_2
FLOOR_3
FLOOR_4
FLOOR_5
```

Users would then include one of these attributes in their SPOOL commands, and the operators would run each of the five environments in turn for short periods throughout the day. For example, a user on the second floor could give the following command:

```
SPOOL MYFILE -ATT FLOOR_2
```

This request would stay in the spool queue until the environment that accepts the FLOOR_2 attribute is started, and locates the request. While that environment is active you know that all the printout must go to the second floor.

To force users to use the system you could make the floor number attribute mandatory in the environment file for the corresponding despooler. In that way, any request that failed to specify a floor number would not be printed. By doing more work, you could set up user-specific or project-specific attributes to add the floor number attribute automatically. See Chapter 3, Defining Printer Environments, for details of how to set up mandatory attributes.

Creating Character Mapping Files

Many PRIME users have printers with character sets that are incompatible with the Extended Character Set (ECS, Prime Standard). This results in the printed files being printed in a format different from the one displayed on a terminal screen.

Character mapping allows you to submit files for printing in one character set, and print them in another. Unless there are no mapping directives in the environment file, character mapping is always attempted. For each printer a default mapping is used, unless you request a different one with the SPOOL command -XLATE option. You implement mappings by using mapping files, each of which describes the mapping from one character set, to the equivalent character output for a printer. See Chapter 3, Defining Printer Environments, for information on the MAP Printer Environment Command.

Note that while a file is being plotted, either by using the SPOOL -PLOT option or by an embedded character sequence in the file, ^001^005, no mapping or formatting will take place.

It should also be noted that mapping can degrade performance. Normally the Rev. 21.0 style of mapping occurs, with characters read assuming an ECS file with space compression and output as ASCII. If you are using QUME printers requiring OAS support you should replace ECS_TO_ASCII_MAP, with QUME_OAS_MAP. Performance is also dependant upon the complexity of the map.

Character Mapping in the SPOOL* Directory

At Rev. 22.0 there are two files in the new directory SPOOL*>MAPPINGS, ECS_TO_ASCII_MAP, and QUME_OAS_MAP. Each file defines a mapping to a despooler, namely a simple map which performs ECS to ASCII mapping, and a map for use with QUME printers requiring OAS support.

You may create a file STANDARD_MAP to hold the default mapping. If the file STANDARD_MAP does not exist when a despooler starts up, then its default mapping is none. With a mapping of none, the characters are read assuming an ECS file with space compression and output as ASCII.

System administrators use the environment definitions to set up the lists of maps available. When a despooler is started it reads in the list of maps set up by the system administrator and constructs a table of available mappings.

The Mapping Process

From Rev. 21.0 the despooler uses literal strings in ECS for the texts inserted during printing. The input and output mappings required, are from the character set of the spooled file to ECS, and then from ECS to the character set of the printer.

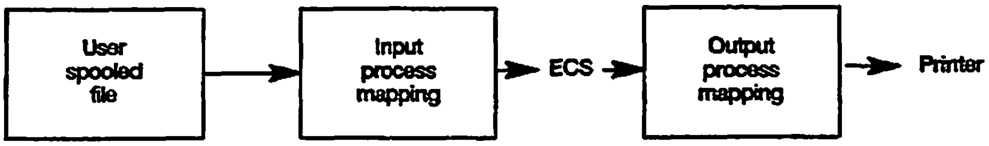

FIGURE 2-3 The Mapping Process

Preparing Mapping Files

A summary describing mapping file creation follows. Greater detail can be found in SPOOL* > INFO > DEFINING_MAPS.RUNO.

You define maps using text files that are referred to as source files. The source file is submitted to a utility SPOOL* > CREATE_MAP.RUN which converts them into structures. Once you have created the map, it is filed in SPOOL* > MAPPINGS so that the despooler can access it.

Map Source Files

Source files consist of comments, keywords and numbers. Comments start with /* and terminate at the end of the line. The numeric values, for example ECS and ASCII, are taken from tables. A single logical line can consist of several physical lines. A physical line which is to continue must be terminated by &. The keywords are listed below.

Keyword	*Abbreviation*	*Explanation*
END		RMAP, REXC, WALTMAP, WMAP and WEXC, introduce a list of values which must be terminated by END.
End_Of_Line	EOL	Defines the character that terminates a line of input.
Form_Feed	FF	Character that starts a new page.
Form_Feed_OFF	FFOFF	Disables testing for form feed character.
Read_ESCape	RESC	Defines the value to be tested to see if special processing is required during read mapping.
Read_MAP	RMAP	Defines table of one-to-one substitutions.
Read_EXCeption	REXC	Defines sequences of characters with instructions for their special processing.

Keyword	*Abbreviation*	*Explanation*
Space_Compress_Off	SCO	Disables the expansion of a space compression character.
SPACing_char	SPAC	Character that moves the print position one character to the right.
Write_ALTernate_MAP	WALTMAP	Defines the alternate write map.
Write_ESCape	WESC	Defines the value to be tested to see if special processing is required during the write mapping.
Write_EXCeption	WEXC	Defines a table of characters that require special processing.
Write_MAP	WMAP	Defines the basic output map of one-to-one substitution

Keywords are placed at the front of a line and may be abbreviated. The keywords RMAP, REXC, WALTMAP, WMAP and WEXC, introduce a list of values that must be terminated by END. You may put keywords more than once in a file, but you may not nest them, nor can there be any other keyword embedded in a list.

Numbers must be in decimal and range from 0 through 255. These numbers describe various character values to the mapping routines. For example, to map the literal EBCDIC zero to ECS zero you would need a table of EBCDIC values, where zero is 0, and a table of ECS values where zero is 176.

Example Maps

Example 1:

Suppose a printer has an escape sequence that sets the line spacing in 216ths of an inch. The format of this command is: ESC 3 n. In ASCII this is 27 51 n, where n is 1 through 255.

When the line spacing feature is used this sequence must not be altered by the spooler system. This sequence must be passed through both the read mapping and the write mapping routines unchanged. To achieve this the following map may be used:

```
REXC
27 51 0 1            /* Pass ESC 3 and next character straight through
                     /* the read routine
END
```

```
WEXC
27 8 51 1               /* Pass ESC 3 and next character straight through
                        /* the output mapping routine.
END
```

Example 2:

Suppose a printer has an eight bit character set. The printer prints the 'o circumflex' character when it receives the character 148. This corresponds to 86 in ECS. Therefore to print the character 'o circumflex' a write map must be defined which maps 86 to 148. For example

```
WMAP
86 148                  /* make ECS o circumflex output the correct symbol.
END
```

Another printer may only have a limited number of characters that do not include the symbol o circumflex. To simulate the o circumflex it is necessary to output o and overprint it with " (double quotation mark). This can be done using the following sequences:

```
WESC 255                /* define the write escape character as something we
                        /* never want to output to the printer.

WMAP
86 255                  /* map the ECS character for o circumflex to the
                        /* write escape.
END

WEXC
86 0 111 8 34           /* instead of one character for o circumflex
                        /* output o backspace ".
END
```

More details about map file creation can be found in the directory called SPOOL* > INFO > DEFINING_MAPS.RUNO.

Setting Up and Controlling Spool Queues

At Rev. 21.0 and later, each system has a single spool queue in the SPOOL_QUEUE* directory. You need take no special action to create the spool queue. Spool requests are automatically added to the local queue when a SPOOL command is issued. The -ON and -DISK options of the SPOOL command let you put requests in the spool queues of other computers in a network. The directories that contain spool queues must not be protected by passwords. You can use command options in an environment file to specify how frequently each spool queue is to be searched by a particular despooler phantom.

Where Files are Stored for Printing

Unless a SPOOL command contains the -NO_COPY or -SWO options, each file to be printed is copied into a SPOOL_DATA* directory. Often, one SPOOL_DATA* directory on the system is sufficient, but if you do a lot of printing it may not be enough. You can have several SPOOL_DATA* directories on a system, although there is still only one spool queue.

Users do not need to know the names of the partitions that hold the SPOOL_DATA* directories. The SPOOL command automatically selects a new data area if the area it is using becomes full.

To use more than one SPOOL_DATA* directory you must

- Create the SPOOL_DATA* directories.
- Set the ACL for each SPOOL_DATA* directory.
- Create a file named DATA_PARTITIONS in SPOOL_QUEUE*.

The DATA_PARTITIONS file consists of a list of the names of the partitions, on the local machine, in which you have created SPOOL_DATA* directories. The partition names are entered without the delimiters < and >.

If possible the SPOOL_QUEUE* directory should not be on the same partition as a SPOOL_DATA* directory, though it can be, if you cannot locate it elsewhere. The drawback to having SPOOL_QUEUE* on the same disk as a SPOOL_DATA* directory is that if enough files are copied to SPOOL_DATA* to fill the disk, the spool queue itself is not able to accept new requests. This is a problem if you have several SPOOL_DATA* directories, but does not matter if you have only one SPOOL_DATA* directory.

When the Spooler subsystem becomes operational, the SPOOL command copies data files to the SPOOL_DATA* directory on the first partition named in the DATA_PARTITIONS file. Data files for printing are copied to that partition until the disk is full, or some other fault prevents the file from writing to that particular partition. Then any partially written entry is removed, and the SPOOL command tries the next partition in the DATA_PARTITIONS list. When the SPOOL command has successfully used a new partition, it continues to use it for each subsequent request until it becomes full, at which point it goes to the next one. The partition is also cycled if a SPOOL -OPEN request is received.

If the SPOOL command encounters difficulty with the last partition in the list, it tries the first partition. If all SPOOL_DATA* partition names are full, the SPOOL command reports an error.

If you do not create a DATA_PARTITIONS file, the SPOOL command locates the first SPOOL_DATA* directory on the system and uses that. You can therefore have a single SPOOL_DATA* directory and operate without a DATA_PARTITIONS file.

You can add new partitions to the list at any time. However, if you want to remove a partition from the list you must use the following procedure:

1. Delete the entry from the DATA_PARTITIONS file.
2. Shut down all the despooler phantoms.

3. Issue the PROP -COLDSTART command. This step ensures that the SPOOL command uses the revised DATA_PARTITIONS file.
4. Start up the despooler phantoms again.
5. Wait until all the files in the SPOOL_DATA* directory that you want to remove have been printed. Use SPOOL -LIST -FULL to check. The full display includes the partition name of the SPOOL_DATA* directory.
6. Delete the SPOOL_DATA* directory.

Viewing the Spool Queues

Users can normally only display their own entries in the spool queue with the SPOOL -LIST command. Privileged users, that is, SYSTEM and members of .SPOOL_ADMINISTRATOR$ can see the entire queue.

To enable all users to see the whole spool queue, you should create a FULL_LIST_USERS file, in the SPOOL_QUEUE* directory, containing the following single line:

```
.ALL_USERS.
```

Note that this line has a period (.) at each end. To enable selected users to view the entire spool queue, you should create a FULL_LIST_USERS file in which each line consists of a single user ID, or a single ACL group name.

To save a FULL_LIST_USERS file you need temporarily to change the ACL for SPOOL_QUEUE* to allow you to write to that directory.

If no FULL_LIST_USERS file exists, it is impossible for any users, except privileged users, to see any requests in the spool queue other than their own.

Creating Your Own Display Spool Queue Routine

You can replace the routine that produces the spool queue display with one of your own, in two ways, allowing you to define your own display for a SPOOL -LIST operation.

- By using the Programmable Interface, which involves the creation of a replacement program. This affects all of the SPOOL -LIST options, -BRief, -DETail, and -FULL or,
- By using the Configurable Listing Interface, an ASCII configuration file applying only to the SPOOL -LIST -BRief option

The Programmable Interface

The existing routine for displaying queue entries has become an entrypoint. The entrypoint supplied by Prime is SP$LST in the SPOOL command library SP$LIB. You can replace the entrypoint by the use of search rules, to invoke your own version of SP$LST. The existing spooler security is not compromised.

The entrypoint takes arguments of the queue entry to be displayed, and the REPORT_OPTIONS structure. Templates of the queue entry and REPORT_OPTIONS structure are provided in insert files, named REP_OPTIONS, stored in SPOOL*>SOURCE. The insert files are in languages PL1, FTN, and C.

The calling sequence is:

```
SP$LST(QE, REPORT_OPTIONS)
```

where:

QE is the entry structure holding the details of an individual request.

REPORT_OPTIONS is a structure which holds the options used for scanning and displaying the queue.

The routine is called once per request to be displayed. If there are ten requests returned from the spool queue then the SP$LST routine is called ten times. It is up to the SP$LST routine to do the testing for such things as user names, matching attributes, style of reports, and so on.

The arguments are both structures. They are both defined in REPORT_OPTIONS.INS.@ in the SPOOL*>SOURCE directory. The only item that should be altered in these passed arguments is the REP_OPTIONS.ENTRY_FOUND flag. If this is still unset when the spooler calling routine finishes scanning the queue, the message `no entries found` is displayed. There are three levels of report; Brief, Detail, and Full. Brief is the default.

The Configurable Listing Interface

You can also configure the SPOOL -LIST output by creating an ASCII file, as described below, and run it through the parser program which reads, deciphers and validates it. The resultant binary file is stored in the directory SPOOL*>CONFIG_LIST and you should name it with your user_id, or project_id (which should be preceded by an asterisk, *), or as the default listing file (.DEFAULT_CONFIG_LIST).

The SPOOL*>CONFIG_LIST directory is then searched for a file which matches your user name, your project_id or which has the name .DEFAULT_CONFIG_LIST. The file, once it is identified, is read by the spool program and the selected items are displayed.

The ASCII file has two sections, the header section (which is optional) and the selection section in which you define the information you want to display and the order in which it is to be displayed.

The example below demonstrates how a user, with the user_id OPS, would configure the display to show how many jobs are in the spool list, their size and the printer to which they have been sent.

```
/* ops.template, spool*>config_list, spooler project, 17/09/90
/* example ascii configuration file for customized display
/* Copyright (c) 1990, Prime Computer, Inc., Natick, MA 01760
/*
/* Header section (optional)
start_header
[Example operator headings]
```

```
Reqno   Size   Copies
Attributes
==================================
end_header
/* the mandatory selection section
start_selection
Request_number   File_size   Copies
Attributes_var 78
end_selection
```

Each section starts and end with a marker, (start_header, end header and start_selection, end_selection). The selection section consists of a list of keywords that defines the information and the order in which it is to be displayed. The file is then processed by the parser program by issuing the following command:

```
R SPOOL*>CREATE_CONFIG_LIST -Input pathname -Output pathname
```

In the case of the above file, OPS.TEMPLATE, the user, who has the user name OPS, will cause an output file, called OPS, to be produced with the following command line:

```
R SPOOL*>CREATE_CONFIG_LIST -I OPS.TEMPLATE -O OPS
```

The file OPS is stored in SPOOL*>CONFIG_LIST so that when the user OPS performs SPOOL -LIST (-BRIEF being the default) the following is displayed:

```
OK, SPOOL -LIST
[SPOOL Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
System SYSA
[Example operator headings]
Reqno   Size   Copies
Attributes
==================================
183        10  5
DOC, PRINT_ROOM
184         1  1
DOC, PRINT_ROOM
185         2  2
DOC, PRINT_ROOM
OK,
```

Configuring Spool Queues

You can instruct a despooler phantom to search queues on other nodes of a network in addition to the local queue by including suitable commands in its environment file. For details see the descriptions of CONFIG, DISK, and NODE in Chapter 3, Defining Printer Environments.

The Rev. 21.0 spooler can search remote queues of both Rev. 21.0 and pre-Rev. 21.0 formats.

For a ring-based network, if you want an environment to deal with all suitable spool requests, no matter which node they are queued on, then you should include a NODE -RING command in the environment file. The despooler then searches spool queues at all the nodes that are defined in the ring configuration.

Defining Queue Polling Rates

You can set several parameters that control the frequency with which queues are checked for work. For details see the descriptions of POLL_RATE, NODE, and DISK in Chapter 3, Defining Printer Environments.

By default, spool queues are searched in a strict order. If an environment is suitable for several of the spool requests in a remote queue, its despooler phantom normally deals with all of them before moving on to search other queues. This means that the local queue can sometimes be delayed for long periods.

You can prevent such delays by including the PRIORITY_TO_LOCAL_QUEUE command in the environment file. This command makes the despooler check the local queue immediately upon finishing with a request from a remote queue.

Subsystem Security

The Spooler subsystem is designed to meet the US Department of Defense C2 classification.

The SPOOL_QUEUE* and SPOOL_DATA* directories are protected by ACLs, and must not be protected by passwords. You must set these to give ALL access to the .SPOOL$$ group and no access to any other users. Users never belong to .SPOOL$$ permanently. The SPOOL_QUEUE* and SPOOL_DATA* directories are manipulated only by the queue management routines. When a user issues a SPOOL command the .SPOOL$$ group is temporarily added to the user_ID only for as long as is necessary to transfer data to the protected directories. This makes it impossible for any user to see or modify files that belong to another user. The SPOOL* directory must also be protected with an ACL. For full details of the correct ACLs for all Spooler subsystem directories and files, see the earlier section of this chapter, Directories and Files Used by the Spooler Subsystem.

Usually a copy of the file to be printed is held in a SPOOL_DATA* directory. For large documents you can use the -NO_COPY option of the SPOOL command. This option prevents the file from being copied into SPOOL_DATA*, and thereby saves disk space. Instead, the original file is copied directly to the printer. It is then the user's responsibility to make sure that the source file is secure. If you use the -NO_COPY option you must give the despooler phantom that will print the file, read access to the file, and give attach access to the directory that holds the file. That is, you must give sufficient access to the name under which the despooler phantom is running

Ordinary users can give the SPOOL command to request that files be printed, to look at their own entries in the spool queue, and to cancel or modify their own print requests. They can also use PROP -STATUS to check which environments are active. However, they cannot view other users' entries in the spool queues, unless the System Administrator deliberately sets up the system to allow it. This might be worth doing if, on your system, it is more important to have some indication of how long it will take to have a file printed than it is to keep the information secret.

All other SPOOL and PROP options are available to privileged users only. To be a privileged user you must be a member of the .SPOOL_ADMINISTRATOR$ group. Do not confuse the .SPOOL_ADMINISTRATOR$ and the .SPOOL$$ ACL groups.

Caution

You must never make any user a member of the .SPOOL$$ ACL group. If you were to put a user in the .SPOOL$$ ACL group the data in SPOOL_QUEUE* and SPOOL_DATA* would not be secure. It is difficult to check who is in which ACL group, because you need to look at each user's profile, so it is best to ensure that you never, under any circumstances, add anyone to the .SPOOL$$ ACL group.

How to Set the Correct ACLs

A utility program is provided that automatically sets the ACLs correctly on all the directories and files on the local disk that are used by the Spooler subsystem. If you have created several SPOOL_DATA* directories then, before you use this utility, you must create a DATA_PARTITIONS file in SPOOL_QUEUE* listing the MFDs that hold the SPOOL_DATA* directories. The utility only sets ACLs for SPOOL_DATA* directories on MFDs listed in the DATA_PARTITIONS file. If there is no DATA_PARTITIONS file, the utility assumes that there is only one SPOOL_DATA* directory, and sets the ACL of the first one that it finds.

To invoke the program, issue the command

```
R SYSTEM>SPOOL_INSTALL_ACL.CPL
```

This utility does not set ACLs on remote disks. You must set ACLs on remote disks by using the PRIMOS commands SAC and EDAC. You may use these PRIMOS commands to set nonstandard ACLs on any file or directory in the Spooler subsystem, but the security of the subsystem cannot be guaranteed if the recommended ACLs are not used.

Compatibility With pre-Rev. 23.0 Spoolers

There were major changes to the Spooler subsystem since Rev. 21.0. If you are upgrading from a pre-Rev. 21.0 Spooler subsystem the System Administrator needs to do some initial work to set up the SPOOL*, SPOOL_QUEUE*, and SPOOL_DATA* directories, to redefine the printer environments and so on, as described in this chapter. However, most of the changes had little or no impact on the visible user interface. Changes that did affect the user interface were:

- Changes to the SPOOL command options
- Changes to the messages that are displayed
- Changes to the screen displays

These changes are described in: Chapter 1, Introducing The Spooler Subsystem; Chapter 4, Routine Operational Procedures; Appendix A, Summary of Spool and Prop Commands; and Appendix C, Spooler Subsystem Messages.

The degree of compatibility is shown in Figure 2–4. The arrows show the direction of data movement.

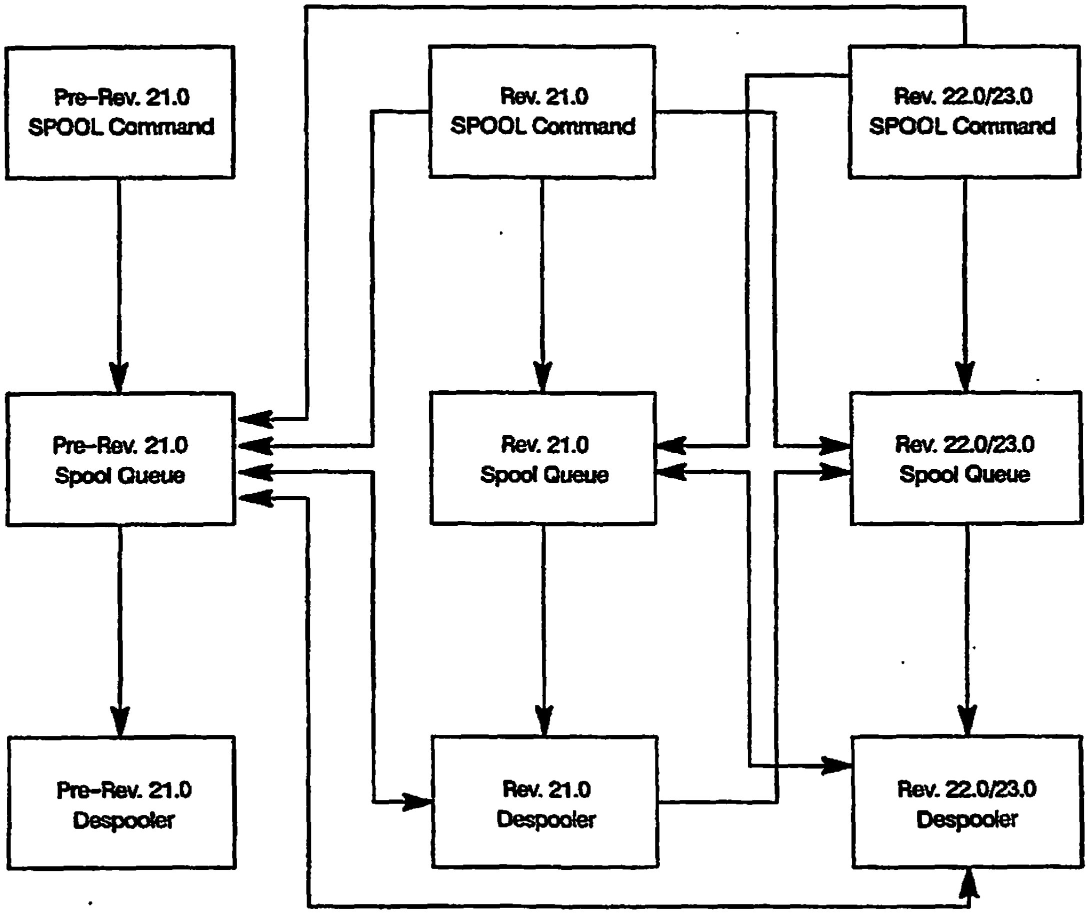

FIGURE 2–4. Compatibility Between Rev. 22/23 and Earlier Spoolers

If you have a network in which some nodes use the Rev. 22 or Rev. 23 Spooler subsystem, and some nodes use earlier revisions, then you should be aware of the following:

- The Spooler subsystem in Rev. 21.0 and higher can add print requests to spool queues on pre–Rev. 21.0 systems, and access the queues for despooling. However, pre–Rev. 21.0 systems cannot access Rev. 21.0 and higher spool queues.
- Users of Rev. 21.0, or a higher system, can direct files to be printed at a printer on any other node in a network. However, where there are pre–Rev. 21.0 Spooler subsystems in the same network, it may be necessary to use the –DISK option to put a request in the correct spool queue.

Print Handlers

A **print handler** is a piece of software, used by a despooler phantom, to perform standard output operations and other related device handling for a particular type of output device. It is not user visible.

Advantages of Print Handlers

The print handler is entirely separate from the despooler. Separated in this way, all output devices appear the same from the viewpoint of a despooler phantom, and the despoolers can use similar output control commands for all the print handlers. Consequently, you can support any new output device by writing a suitable print handler. There is no need to make any change to the despooler code.

Print Handlers Provided by Prime

Print handlers are provided for the following devices:

DPTX
POSTSCRIPT
NEC
QMS
QUME
TTY
DATAPRODUCTS
DIABLO
PRINTRONIX

The table below summarizes the type of printer, and the line protocol normally associated with that print handler.

Print Handler	*Type of Printer*	*Line Protocol*
DPTX.PLP	DPTX	DPTX
POSTSCRIPT.PLP	PostScript	Typically a serial line using XON/XOFF, no echo
NEC_OAS.PLP QUME_OAS.PLP	OAS_PRINTING using ETX/ACK protocol	Serial line but not using XON/XOFF
TTY.PLP	Transparent but recognizes EVFU skips	Parallel or serial
DIABLO.PLP	Parallel or serial	Async, XON/XOFF
NEC.PLP		Async, XON/XOFF
QUME.PLP		Async, XON/XOFF
DATAPRODUCTS.PLP		Parallel line
PRINTRONIX.PLP		Parallel line

The code for the print handlers is located in the file SPOOL*>DESPOOL_LIBRARY.RUN, which is a library class EPF.

The following notes give broad guidance on the selection of which print handler to use, with which printer and communication medium.

> The PostScript and DPTX printers each have their own special print handler which should be used.
>
> If a printer is supplied with a print handler, that print handler should be used.
>
> The communication medium used (serial, parallel, PDN, TCP/IP) can give guidance on which print handler to use.
>
> If a parallel communication medium is used, try the following print handlers: TTY, DataProducts, or Printronix.
>
> If a serial communication medium with XON/XOFF is used, try the following print handlers: NEC, QUME, or DIABLO.
>
> If a serial communication medium with ETX/ACK is used, try the following print handlers: NEC_OAS or QUME_OAS.

Specifying Which Print Handler to Use

To specify the print handler for a despooler, use the DEVICE command in the environment file, as explained in Chapter 3, Defining Printer Environments.

Appendix B, How To Write A Print Handler, gives full details of how to create additional print handlers for devices that are not directly supported by Prime.

Control of Laser Printers by PostScript Procedures

PostScript is a high-level language used by intelligent printers to format their output. There are three ways in which you can use the PostScript facilities.

- Using the SPOOL command option -NO_FORMAT, spool a file containing PostScript commands, so that the printer directly obeys the commands.
- Using the SPOOL command option -PROC *name*, you can instruct the printer to process the file with PostScript procedure *name*.
- Using the SPOOL command with neither the -NO_FORMAT nor -PROC options, the PostScript despooler will include PostScript commands with each line that cause the lines to print.

To use a PostScript printer you must give a SPOOL command that specifies the correct attributes to select the PostScript printer.

To use a PostScript procedure you must also use the -PROC option to supply the name of the PostScript procedure.

PostScript dictionaries are first loaded when a printer's despooler phantom is started.

For instructions on how to write PostScript procedures you should consult the manual supplied with the printer, or a standard textbook such as the following, published by Addison-Wesley Publishing Co.:

PostScript Language Reference Manual, ISBN 0-201-10174-2,
PostScript Language Tutorial and Cookbook, ISBN 0-201-10179-3.

To set up your PostScript printer to use PostScript procedures, follow the steps below.

1. Store your PostScript procedures in files on the system.
2. Create a file that contains the pathnames of the PostScript procedure files. The file POSTSCRIPT_INIT in the SPOOL* directory is supplied for this purpose, though you are free to create other files if you wish.
3. Specify the file that you created at step 2 in a printer environment file with a DEVICE -DICT command. If you do not specify a pathname, then SPOOL* > POSTSCRIPT_INIT is used by default. See Chapter 3, Defining Printer Environments for details.
4. Make sure Serial PostScript printers have the -NO_ECHO option on the ASYNC lines of their .ENV files.

You can create several files that refer to PostScript procedures, and specify each in a different environment file. It is therefore possible to have several PostScript printers on the system.

Cold Start Procedure

At cold start, the Spooler subsystem must be started after the system date and time are set.

The startup procedure consists of the following three steps:

1. Share the Spooler subsystem data area.
2. Issue the PROP -COLDSTART command.
3. Start up despooler phantoms (optional).

The command input file for system startup (PRIMOS.COMI) that is supplied with the delivered system performs the first two tasks of this sequence. The first two tasks are needed even if no despooler phantoms are run. The PRIMOS.COMI file contains the following command:

```
COMI SYSTEM>SPOOL.SHARE.COMI
```

The file SYSTEM>SPOOL.SHARE.COMI, in turn, contains (among others) the following command:

```
PROP -COLDSTART
```

The PROP -COLDSTART command initializes all the Spooler subsystem data structures and resets the queue control information. You should never need to issue this command except as part of the system cold-start routine, though members of the .SPOOL_ADMINISTRATOR$ ACL group can, if necessary, issue the command at any time. That need might arise after you have removed a SPOOL_DATA* directory or after some problems with a despooler phantom.

The following describes what PROP -COLDSTART does in particular circumstances.

Circumstance	*Result*
with no despoolers running and queues empty	shared segment and request numbers are all re-initialized.
with entries in the queues	the printing flags are reset, but the count is not. The shared segment is re-initialized.
with active despoolers	a question and answer dialog asks if you wish to shut the despoolers down. Reply YES to shut each one down. The circumstances and results described in the first two items then apply.
with active despoolers some environments not shut down	the entries for environments still running are left in the shared segment, the entries for ones to be shut down (if any) are removed. The spool queue is not re-initialized. Rogue entries, for despoolers that have logged out but are still listed in the shared segment, are automatically removed by a PROP -COLD.

Despooler Phantoms and System Cold Start

To start despooler phantoms as part of the system cold-start procedure, you must insert into SYSTEM>SPOOL.SHARE.COMI the PROP commands that start up the environments that you want to make available. For example:

```
PROP DOC -START
PROP LQP -START
```

The insertion point for PROP commands is marked in SYSTEM>SPOOL.SHARE.COMI.

For instructions on how to start, stop and suspend phantoms at any other time see Chapter 4, Routine Operational Procedures.

Setting Up the Electronic Vertical Format Unit (EVFU)

Some printers use an Electronic Vertical Format Unit (EVFU) to print forms quickly and efficiently. A printer with an EVFU is typically used for printing invoices, salary slips, and similar output where a few items of information have to be accurately placed on particular lines of a form. The files that are printed must include special control codes to operate the EVFU.

When you create an environment file for a printer that uses an EVFU you must do three things, as follows:

- Create an EVFU file.
- Use the -EVFU option of the DEVICE command in the printer environment files to specify the pathname of the EVFU file.
- Ensure that the parameters in the FORMAT command of the environment file are consistent with the EVFU file.

Note

> If a printer that uses an EVFU either loses power or is switched off, then the paper must be realigned correctly and the Despooler phantom that controls the printer must be stopped and restarted to reload the EVFU memory.

EVFU Files

An EVFU file assigns channel numbers to designated physical lines on the form. All EVFU files must conform to the following rules:

- Each line of the form can only be associated with one channel.
- The first line of the form must always be represented by channel 1.
- A maximum of 12 channels can be designated.
- Only channel 12 can be assigned to more than one line.
- The maximum form length for a 300 lpm printer/plotter is 132 lines.
- The maximum form length for a band printer is 143 lines.
- The EVFU file must reside in the SPOOL* directory.

The files themselves are ASCII files, created with an editor such as ED or EMACS.

An EVFU file can be in either the new format or the old format. Both formats are described below.

You should use the new format for all new EVFU files. The new format allows you to specify directly the line numbers associated with channels, and makes it easy to construct correct EVFU files.

In old format EVFU files, channel numbers are on the exact lines that they control. This format is difficult to read and maintain because of the large number of blank lines.

This section uses an example to explain EVFU files and to compare new and old format files. The example specifies four channels, with channel 1 associated with line 1 of the form, channel 2 with line 5, channel 4 with line 14, and channel 10 with line 40.

```
Channel 1 Line 1

Channel 2 Line 5

Channel 4 Line 14

Channel 10 Line 40
```

FIGURE 2–5. New Format for EVFU File

New–format EVFU Files: In the new format, line and channel numbers are directly specified as shown in Figure 2–5.

Old–format EVFU Files: Old format EVFU files are graphic representations of the forms, or page layouts, that they are intended to format. You must ensure that the number of lines in an old–format EVFU file is equal to the number of lines in the form.

The EVFU file assigns **channel numbers** to designated physical lines on the form. For example, if you want channel 5 to be on line 20, put the number 5 on the 20th line of the form. (If you have ever created a paper tape control loop for a printer, it may help you to think of the channels in the EVFU file as corresponding to the columns on the paper tape loop.)

Figure 2–6 shows how to use ED to create and file an old format EVFU file. The Figure 2–6 file conveys exactly the same information as the new format example of Figure 2–5.

EVFU files in the old format are subject to the following additional restrictions:

- The channel number must be the first non–blank character on the line.
- The first line of the form is always represented by channel 1. Thus, 1 must appear on the first line of the EVFU file.
- Lines without channels assigned to them can contain a 0, or can be left blank.

Note that if you wish, you can enter comments on any line that contains a channel number.

```
OK, ED
INPUT
1



2








4
























10
EDIT
FILE PAYROLL-FORMAT
OK,
```

FIGURE 2-6. Using ED to Create an Old Format EVFU File

How to Use an EVFU File

When you use EVFUs, you embed control characters in the text file that is being formatted for printing by the EVFU file. The control characters are skip to channel instructions. When the printer receives such an instruction it skips to the line designated as that channel in the EVFU file.

Figure 2-7 shows a file that is to be formatted by either of the EVFU files shown above.

```
^001^000                                   /*Enter NO_FORMAT print mode
^003^001  Top  of  page  is  here          /*Channel 1
^003^002  This  is  Line  5.........       /*Channel 2
  Line  6  information  is  here.......    /*More output file lines
  Line  7.............                     /* are put in here
^003^004This  is  on  Line  14.......      /*Channel 4
  Line  15............                     /*Another output file line
^003^012This  is  on  Line  40.........    /*Channel 10 in octal
  Line  41............                     /*Another output file line
```

FIGURE 2-7. Form to be Formatted by the EVFU

Comments are prefaced with the /* marker. Comments would not ordinarily be present but have been included in the above example to clarify what is in the file.

The control character ^003, which must be in column 1, tells the despooler phantom to look at the next character, which gives the number (in octal) of the channel to skip to. In this case, the phantom is to skip to channel 1 (top of page), channel 2 (Line 5), channel 4 (Line 14), and channel 10 (Line 40). When the file is spooled the normal despooler format control must be disabled. Formatting can be disabled by one of the following two methods:

- You can use the -NO_FORMAT option of the SPOOL command.
- The file can have the control characters ^001^000 as the first line.

Embedded Control Codes in Printable Files

The following is a list of control codes which, when embedded in a file, are interpreted by the Spooler. Control Codes must start in column 1 unless otherwise stated. Control codes do not print anything after the code sequence unless otherwise specified. Channel (both vertical and horizontal) commands apply to the logical page.

Change of Mode Commands

Control Character(s)	*Interpretation*
^001^000	Enter NO_FORMAT print mode
^201^000	Entering NO_FORMAT mode puts the driver in **pass through** mode where all text is passed to the device and is not interpreted by the device driver.

^001^001 ^201^001	Enter Fortran Control Mode. Spooler now interprets: 1 + 0 – Standard Fortran codes that appear in column 1.
^001^002 ^201^002	Enter COBOL Control Mode. Identical to FORTRAN mode except the first two characters (format characters) are used. The first character is the same as in FORTRAN format, the second character is always a space.
^001^003 ^201^003	Enter Paginate mode (driver default). When in this mode, the driver interprets a ^014 or a ^214 anywhere as a form feed. A page eject is performed automatically when BM is reached. A page heading is printed at the top of each page provided TM is greater than 1.
^001^004 ^201^004	Enter No header mode. Same as paginate mode except that no page headings are printed.
^001^005 *nnn* ^201^005	Enter Raster plot mode. Binary where *nnn* = number of words to dump.
^001^036 *header* ^201^036 *header*	New Header Line – DO NOT reset page count. Enters paginate mode if not already in it. Where *header* is the new header in ASCII. It causes a page eject.
^001^037^*nnn*^*nnn* ^201^037^*nnn*^*nnn*	Enter new page size specified by the 16–bit number contained in the two *nnn* bytes. Obsolete option, maps into set bottom margin.
^000^001 *header*	New Header Line – DO reset new page count.
^001 *header* ^201 *header*	Enters paginate mode if not already in it. Where *header* is the new header in ASCII. Causes a page eject.
^003^*nnn*	EVFU channel skips the channel to skip to must be in the range: 1 >= channel number >= 12 If *nnn* is not given then the channel number defaults to 1, which is implied Top–of–Form and advance TM (channel 1).

Logical Page Formatting Commands

Control Character(s)	*Interpretation*
^002^001^*nnn*	Left margin, where *nnn* is the column number of the left margin in ASCII.
^002^002^*nnn*	Right margin, where *nnn* is the column number of the right margin in ASCII.

^002^003^*nnn*	Top margin, where *nnn* is the line number of the top margin in ASCII.
^002^004^*nnn*	Bottom margin, where *nnn* is the line number of the bottom margin in ASCII. Attempting to print beyond this causes a page eject.
^002^005	Wrap.
^002^006	Truncate.

Form Formatting Commands (Reserved)

Control Character(s)	*Interpretation*
^002^010^*nnn*	Lines per inch, where *nnn* is the number of lines per inch in ASCII. The physical page is also changed to accomodate lines per inch.
^002^011^*nnn*	Characters per inch, where *nnn* is the number of characters per inch in ASCII.
^002^012^*nnn*	Set the physical height in lines in ASCII.
^002^013^*nnn*	Set the physical width in columns in ASCII.

Horizontal Movement Commands

Control Character(s)	*Interpretation*
^002^020^*nnn*	Set the tab interval, where *nnn* is the tab interval in ASCII.
^002^021^*nnn(list)*	Set tabs, where *nnn* is a tab stop in ASCII, and *list* is of additional tab stops in the form of ^nnn^nnn^nnn ... to a maximum of 128.
^002^022^	Clear tabs.
^002^023^*nnn*^*mmm*	Set channel skips, where *nnn* is the channel number and *mmm* is the line to skip to, both in ASCII.
^002^024^	Clear VFU. Sets channel 1 to line 1 and clears all other VFU settings.

Other Controls

Control Character(s)	*Interpretation*
^002^030	Look for boldface commands on this line.
^201	Bold overstrike on, this may appear anywhere on the line.
^202 or <CR>	Bold overstrike off, this may appear anywhere on the line.
^002^031	Look for underscore commands on this line.
^201	Underscore on. this may appear anywhere on the line.
^202 or <CR>	Underscore off. this may appear anywhere on the line.
^002^032	Covert lower case to upper case before printing.
^002^033	Print lower case.
^002^034	Print line numbers in left margin.
^002^035	Do not print line numbers.
^002^036^*nnn*<*text*>	Skip *nnn* lines and print *text* which is an ASCII text string.
^211 <HT>	When embedded in page mode, tabs to next tab stop.
^212 <LF>	End of line terminator. Causes a <CR><LF> to be output to the device in page mode.
^213 <VT>	Skip to next VFU channel, or 4 lines if not set.
^214 <FF>	A form feed and skip of TM is done when embedded in page mode.
^215 <CR>	End of line terminator. Causes a <CR> to be sent to the device when in page mode.

Control Code information can also be found in the Subroutines Guide, Volume IV.

The Accounting Interface

Each despooler records a line of basic accounting information in its log file on completion of each print job, provided that the log file is active. This information may be adequate for your uses. However, if you need to keep more sophisticated accounts, an experienced programmer can create an accounting routine to your exact requirements. The routine that you create must be an EPF library subroutine with the entrypoint SP$ACT. You must also alter the search rules in SPOOL*>ENTRY$.SR so that the new EPF can be found.

Data Structure Information

The call to SP$ACT takes two arguments. The first is a data structure that matches the PL1 declaration shown in Figure 2-8. All character fields are fixed length to make the structure compatible with FTN. Insert files are supplied in SPOOL*>SOURCE for PL1 and FTN.

The second argument to SP$ACT is a four-element array, each element being a fixed-length string of 80 characters. The SP$ACT routine can set this structure to hold four lines of user-supplied information, that is then appended to the standard trailer page (if a trailer page is used).

If the open_option_used bit is set, then the filename and dtm fields are not significant.

The PL1 data declaration set is shown in Figure 2-8.

```
dcl 1 accounting_data,
2 version bin,
  2 environment_name char(16),                   /* Spooler environment name */
  2 origin_node char(32),                        /* Requesting user's node name */
  2 user_name char(32)                           /* Requesting user's login name */
  2 user_no bin,                                 /* Requesting user's user number */
  2 project_id char(32),                         /* Requesting user's login project */
  2 request_time char(12),                       /* Time of request, yymmddhhmmss */
  2 file_name char(160),                         /* Full path name of file to print */
  2 file_size bin,                               /* Size in records * no of copies */
  2 no_of_copies bin,                            /* No of copies requested */
  2 dtm bin(31),                                 /* File system format */
  2 no_of_attributes bin,                        /* No of names following */
  2 attribute(32) char(16),                      /* Attribute names from request */
  2 options,
    3 no_copy bit(1),                            /* -NOCOPY option used */
    3 numbers bit(1),                            /* Generate line numbers */
    3 no_header bit(1),                          /* No header page */
    3 no_final_eject bit(1),                     /* Suppress final page eject */
    3 no_format bit(1),                          /* No format (-NOFMT) */
    3 fortran_format bit(1),                     /* Fortran format codes */
    3 plot_file bit(1),                          /* Plot file */
    3 notify_user bit(1),                        /* Notify user on completion */
    3 truncate bit(1),                           /* Truncate long lines */
    3 cobol_format bit(1),                       /* -COBOL */
    3 no_overprint bit(1),                       /* -NOP */
    3 open_option_used bit(1),                   /* Set if -OPEN or SPOOL$ equiv */
    3 no_header_format bit(1),                   /* -NPH */
    3 suppress_file_info bit(1),                 /* -SFI */
    3 from_page bit(1),                          /* -FROM */
    3 to_page bit(1),                            /* -TO */
  2 options2,
    3 swo bit(1),                                /* -SPOOL_WHILE_OPEN */
    3 xlate bit(1),                              /* -XLATE */
    3 rfu bit(14),                               /* RFU as spare options */
  2 raster_size bin,                             /* Words per raster in plot file */
  2 start_print_time char(12),                   /* Print started at yymmddhhmmss */
  2 end_print_time char(12),                     /* Print completed at yymmddhhmmss */
  2 terminating_status bin,                      /* See below */
  2 no_of_backups bin,                           /* No of times print backed up */
  2 no_of_restarts bin,                          /* No of print restarts */
  2 total_pages bin(31),                         /* Total pages including restarts */
  2 total_lines bin(31),                         /* Total lines including restarts */
  2 total_characters bin(31),                    /* Total chars including restarts */
  2 from_page_num bin(31),                       /* from page number */
  2 to_page_num bin(31),                         /* to page number */
  2 map_name char(32);                           /* name of map */
%replace accounting_version_ by 2               /* Value in VERSION above */
%replace SUCCESS_ by 0,                         /* Terminating status values */
    ABORTED_ by 1,
    DROPPED_ by 2;
```

FIGURE 2-8. PL1 Data Declaration for SP$ACT

Coding Rules

The SP$ACT accounting routine can perform any actions that do not destroy the despooler program from which it is called. The rules for writing the routine are as follows:

- The accounting routine can attach to a new directory and can safely leave the despooler phantom attached there on return.
- Files opened by the accounting routine must be opened using the K$GETU key, and must be closed before returning to the despooler.
- The accounting routine must not attempt to drive the printer as this can cause undesirable effects when printing continues after return to the despooler.
- The accounting package can invoke EPF or static mode programs so long as they do not destroy the despooler which is itself an EPF.
- Any dynamically allocated memory must be released before returning control to the despooler

Note

> Any conditions signaled while in the accounting routine, and not handled by it, cause the despooler to shut down.

Calling the Accounting Routine

When you have coded and tested your accounting routine you should uses BIND to create a program class library EPF, and add it to the search rules in SPOOL*.

When a despooler phantom completes its first print job it checks for the existence of SP$ACT. If an SP$ACT is found, it is run after each print job. If it is not found at the first attempt, no further attempts are made to call it.

Printout Distribution Addresses

To help with distribution of printout, you can arrange for distribution information, such as the office number to which a listing should be sent, to be printed at the top right of the banner page of each printed file.

How to Set up Distribution Addresses

To set up distribution addresses you must create a file in SPOOL* with the name USER_ADDRESSES. Each line of USER_ADDRESSES must contain a user-ID and the distribution information for that user, with a space or spaces separating the two items. The address information can be a maximum of 32 characters.

If you have several computers in a network, and you make use of this feature, you will have to give some thought to the best method of keeping the USER_ADDRESSES files up to date. The simplest method is to keep identical files on each node, irrespective of who is able to log in on that node, but if you have many users the time taken in searching the file could be unacceptable. The routine that is supplied uses a simple sequential search. You could implement a method of searching that suits your organization and methods of file organization by providing your own routine as described in the next section.

Using Your Own Address Distribution Routine

A routine named SP$ADDRESS is supplied as an EPF library routine in the SPOOL command library (SP$LIB). The SPOOL command calls SP$ADDRESS when a request is submitted to a Rev. 21.0 queue, or when a request in a Rev. 21.0 queue is modified.

The SPOOL command software passes the user-ID to the SP$ADDRESS routine, or if the -ALIAS option was used in the SPOOL command, it passes the alias instead. The SP$ADDRESS routine then consults the file SPOOL*>USER_ADDRESSES, and returns the address information to the SPOOL command software.

You can supply a routine of your own that is to be used in place of the existing SP$ADDRESS routine. The call to SP$ADDRESS in the SPOOL command is as follows:

```
call sp$address(user_name, user_address)
```

Both user_name and user_address are 32-character fixed strings. The user_address parameter is preset to a string of 32 space-characters, so that if SP$ADDRESS has no information to return it should leave user_address unchanged.

When you have coded and tested your new routine you should use BIND to create a program class library EPF, and alter the search rules in SPOOL*>ENTRY$.SR so that your new routine is located before that in SP$LIB.

Banner Page

The standard banner page gives the following information:

- User ID
- Request number and node from which the request was made
- User name and distribution address
- User ID or alias in large letters (See -ALIAS option of SPOOL command.)
- Filename or replacement name in large letters (See -AS option of SPOOL command.)
- Copyright message
- Despooler environment used
- Print request attributes
- SPOOL command options given
- File size in records
- File pathname

- Time the file was last modified
- Time the request was made, and time that printing started
- Message specified in the environment file

Using Your Own Banner Page Routine

You can replace the routine that produces the standard banner page with .one of your own. The method is explained below. The existing routine is a library class EPF with entrypoint SP$HDG in SPOOL*>DESPOOL_LIBRARY.RUN. The routine that you create must also be a library class EPF, named SP$HDG. You are free to put your routine anywhere you wish in the file system, but you must modify the search rules in SPOOL*>ENTRY$.SR so that your replacement for SP$HDG is found before the supplied version. For convenience you should put any replacement routine in SPOOL*, and modify the search rules list to read as follows:

```
spool*>sp$hdg.run
spool*>despool_library.run
```

Data Structure Information

The banner page routine is called by the PRINT_FILE routine of the despooler phantom. PRINT_FILE passes the arguments listed below, and these must be declared in your banner page routine.

```
call SP$HDG (start_date_time, copyright_line, vcom, envcom, pagcom, RBI, envname)
```

```
start_date_time char(15)
copyright_line char(78) var
qcom structure
envcom structure
pagcom structure
rolling_banner_index bin
environment_name char(16) var
```

You can use the information passed in these data structures but you should not alter any of them. The qcom, envcom and pagcom structures are given in full in Appendix B, How to Write a Print Handler.

Coding Recommendations

The source code of the banner page routine that Prime supply is available in SPOOL*>SOURCE>SP$HDG.PLP. You should use this as a template for your own banner page routine. If the existing banner page does not exactly meet your needs, you can produce the banner that you want with a modified version of the existing routine.

Multi-language Despoolers

Sites that use several languages, through the Software Internationalization Tools, (SIT), can have despoolers that run in several languages. These languages are handled through the -ENV_LANGUAGE (-ELANG) option to PROP. This option defines the language used by the environment file. SIT support for the language is needed for -ELANG to work correctly.

The -ELANG option can be used with the -VERIFY and -START option to PROP. For example, to start a French despooler on an English (default language) site, the command format would be:

```
PROP <env> -START -ELANG FRENCH
```

The command to validate the despooler environment would be:

```
PROP <env> -VFY -ELANG FRENCH
```

where FRENCH is a SIT supported language.

3

DEFINING PRINTER ENVIRONMENTS

Introduction

This chapter explains how to use environment files to define printer environments. The commands that can appear in environment files are arranged in five groups, each group dealing with a different aspect of the environment. The chapter explains fully how each command is used. The information is intended for the System Administrator, or other person with responsibility for setting up and maintaining the Spooler subsystem.

What an Environment File Is

Environment files define printer environments. All the information about how the despooler phantoms control the printers available on the system is located in environment files. This information is not narrowly restricted to technical features of the printer, but also deals with

- Recognizing acceptable entries in a spool queue
- Which spool queues to search
- How frequently to search each queue
- Some aspects of formatting
- Logging despooler activity
- Other information that the despooler needs.

For a general explanation of printer environments see Chapter 2, Setting Up The Spooler Subsystem.

Figure 3-1 shows a typical environment file. The comments, each prefaced by a marker (/*), explain the purpose of each line. These particular comments would probably not be included in an environment file as simple as this example, but more realistic comments can be included in the same way. Comments can either explain difficult code or give administrative information such as original coding date, modification dates, and programmer's names.

```
/* QUME.ENV A. H. Acker 06/19/90.
/* This contains the environment definition for the QUME Printer.
/*
ASYNC  -LINE 4 -PROTOCOL TTY -CD 200 -XOFF
                      /* Sets the line configuration.
                      /* Default values except for protocol.
DEVICE QUME -EVFU_OFF /* Specifies name of device Driver.
                      /* No EVFU so software emulation
                      /* of EVFU is needed.
FORMAT -W 80 -L 66    /* Set number of columns and lines
                      /* on the page.
MESSAGE This is the QUME printer on SYSA.
                      /* Define message to appear on each
                      /* page of the header
LOG -ON               /* Log details of current despooler only.
                      /* Don't keep details of earlier
                      /* invocations.
ATTRIBUTE QUME -MAN   /* Accept only requests that include
                      /* QUME as an attribute.
ATTRIBUTE SYSA        /* Accept requests for SYSA machine.
ATTRIBUTE SITE_1      /* Printer located at SITE_1.
CONFIG QUEUES         /* Refer to SPOOL*>QUEUES for list of Nodes
                      /* and Disks where spool queues are located
HEADER 1 -SAME        /* Define number of header pages, and which
                      /* symbols to make large characters from.
TRAILER -ON           /* Print standard trailer.
WARNING -ON           /* Give notification on the printer itself
                      /* when a job is dropped, despooler phantom
                      /* stopped or started, or similar
                      /* major event occurs
```

FIGURE 3-1. Example of an Environment File

How Environment Files are Used

When a despooler phantom is started, either at system cold start, or by an operator, the information in the corresponding environment file is copied into the despooler's own data area. The despooler phantom finds any spool requests in the spool queues specified in the environment file, finds the print requests that it can handle, and arranges to print the files.

Note that each spool request includes its specific printing requirements, such as the type of paper and the site at which the file is to be printed. The file is printed by the first available despooler that meets the requirements.

The spool request uses **attributes** to tell the despooler phantom of its print requirements. The attributes in the SPOOL command must be possessed by any environment that prints the file, as explained in Chapter 2, Setting Up the Spooler Subsystem.

A despooler phantom accepts a spool request if it meets the following criteria:

- All the attributes in the spool request are also attributes of the printer environment, that is, are specified in the environment file.

- The spool request includes all the attributes that are specified in the printer environment with the ATTRIBUTE -MANDATORY command.
- The file size is within the permitted range.

If a user does not specify any attributes in the SPOOL command, or omits some essential attributes, special attributes files can supply default attributes. However, you must have previously created the files of default attributes, as explained in Chapter 2, Setting Up the Spooler Subsystem.

A despooler phantom controls one, and only one, output device. A despooler phantom cannot serve more than one output device. Each active despooler phantom must serve a different output device. There cannot be two or more active despoolers that send output to the same device.

While each despooler phantom is running it uses its own copy of its environment data. This means that you can freely modify an environment file while a despooler phantom is running, without affecting the way the despooler operates. Environment files are consulted by the Spooler subsystem when a despooler phantom is started or reset, and when the environment is displayed or verified.

Creating Environment Files

Environment files are ASCII text files. They must be located in the SPOOL* directory. You can create and edit them with any standard text editor, for example ED or EMACS. For instruction on how to use ED, see the *Prime User's Guide.*

Naming Environment Files

You must use the following convention for names of environment files:

```
environment.ENV
```

where *environment* is the environment name (maximum 16 characters).

When a despooler phantom is started from the supervisor terminal, it takes its user name from the environment filename. Consequently, any despooler phantom that is started as part of the system cold start takes its name from the environment filename. A despooler started from a user's terminal takes the name of the user that started it.

It is good practice to choose meaningful names for environments, for example, LQP for an environment that includes the system's Letter Quality Printer, or DOC for a line printer set up with document paper. The name that you choose must be a valid PRIMOS filename. For example the name must not begin with a numeral.

Environment File Format and General Restrictions

An environment file is a list of commands. Each line in the file can contain a maximum of 128 characters. If you type more than 128 characters in a line, the line is truncated. If you need more than 128 characters to specify a feature of the environment, you can continue on succeeding lines. To show that the next line is a continuation, you must type an ampersand (&) symbol as the last character on a line. A command can take a maximum of eight lines.

When you have prepared and filed an environment, give the following command:

```
PROP environment -VERIFY
```

The syntax of the file is checked by the PROP command software, and you are notified of any errors.

Summary of Printer Environment Commands

In this chapter the available commands are in five groups according to function. The groups are as follows:

- Scheduling print jobs
- Configuring the queue search mechanism
- Formatting the printer output
- Configuring the data line to the printer
- Logging spooler activity

The commands are given alphabetically in the index to this manual under the entry **Environment commands.**

Commands can be typed in upper or lower case. Several commands refer to special files and directories that are used by the Spooler subsystem. For a summary of the directories and their contents, see Chapter 2, Setting Up The Spooler Subsystem. Unless otherwise stated, all numeric arguments and parameters are in decimal.

Scheduling Print Jobs

Commands in this group allow you to make the despooler recognize which of the jobs in the spool queue it is able to deal with, and to decide which files to print first.

▶ ATTRIBUTE name [-MANDATORY]

Where *name* is a printer/document attribute with a maximum of 16 characters. -MANDATORY means that only spool requests with that attribute are accepted.

An environment file can contain a maximum of 32 ATTRIBUTE commands. See Chapter 2, SETTING UP THE SPOOLER SUBSYSTEM, for a full discussion of attributes.

▶ $\begin{Bmatrix}\text{MAX\_SIZE}\\ \text{MAX}\end{Bmatrix}$ n

Specifies the maximum size of file that the despooler accepts. *n* is the number of records. If MAX_SIZE is omitted there is no limit on the file size.

▶ $\begin{Bmatrix}\text{MIN\_SIZE}\\ \text{MIN}\end{Bmatrix}$ n

Specifies the minimum size of file that the despooler accepts. *n* is the number of records. If MIN_SIZE is omitted, the default value of 0 (no limit) is used.

▶ $\begin{Bmatrix}\text{PRIORITY\_SIZE}\\ \text{PS}\end{Bmatrix}$ n

Gives priority to spool requests for files that are shorter than *n* records.

Caution

If you use this option, the despooler phantom uses a more complex method to search the spool queue, and gives reduced overall performance. You should use this option only when it is important that small files should be printed with minimum delay.

Configuring the Queue Search Mechanism

The commands in this section allow you to tell the despooler phantom which spool queues it is to use, and how frequently it should search them for work.

If you need the best possible performance, you can put explicit NODE and DISK commands in the environment files, so that each despooler phantom knows exactly where the queues that it must search are located. If you need to set up several despoolers that use the same queues you can put the NODE and DISK commands in a separate file and refer to that file with a single CONFIG command in each environment file.

Alternatively, you can allow each despooler to find for itself all of the spool queues in the local ring, by giving the NODE -RING command in each environment file. This approach involves two extra overheads: an extra overhead during initialization, because the despooler searches all the available disks for spool queues; and another overhead each time the despooler checks to see whether any more disks have been added that contain spool queues. You would use this approach if it does not matter that each despooler phantom searches every available spool queue, and that system resources aren't used as efficiently as possible. Note, -RING does not include the local machine, only other nodes. If you want to include the local machine use NODE -LOCAL as well.

The total combined number of NODE and DISK commands, including any that are automatically generated, must not exceed 100. For example, specifying a node as pre-Rev. 21.0 generates DISK commands for all added disks on that node. Similarly the NODE -RING command generates a NODE command for each node on the local ring. You can use the -VERIFY option of the PROP command to check that you have not exceeded the limit.

If you need to give the same queue search commands to several despooler phantoms, you can put the commands in a file in SPOOL*, and give a reference to the file with a CONFIG command in each environment file.

Usually the file SPOOL*>QUEUES is used for this purpose: do not confuse this file with SPOOL_QUEUE*>QUEUE, which is the spool queue itself.

▶ CONFIG filename

Specifies where common spool queue commands are located. *filename* is the name of a file in the SPOOL* directory.

▶ DISK name [options]

Specifies the name of a partition that holds a spool queue of pre-Rev. 21.0 format and states how frequently the despooler should poll the spool queue to find work. You can have none, one, or several DISK commands in an environment, provided that the combined number of DISK and NODE commands does not exceed 100. *name* is the name of a disk partition that holds a spool queue of pre-Rev. 21.0 spooler format.

The options determine how frequently the despooler polls each spool queue to find work, as explained below.

Options	*Description*
$\left\{\begin{array}{l}\text{-FAST\_INTERVAL}\\ \text{-FI}\end{array}\right\}$ n	Fast poll rate. *n* is the interval in minutes.The default is 1 minute.
$\left\{\begin{array}{l}\text{-SLOW\_INTERVAL}\\ \text{-SI}\end{array}\right\}$ n	Slow poll rate. *n* is the interval in minutes. The default interval is 5 minutes.
$\left\{\begin{array}{l}\text{-QUEUE\_SEARCH\_INTERVAL}\\ \text{-QSI}\end{array}\right\}$ n	Length of time to wait before retrying an unavailable spool queue. *n* is the length of time in minutes. The default is 10 minutes.
$\left\{\begin{array}{l}\text{-DECAY}\\ \text{-DCY}\end{array}\right\}$ n	Number of times to poll a spool queue unsuccessfully at the fast rate before changing to the slow rate. The default is 10 attempts.

The despooler uses the fast poll rate as long as it finds work each time. If on successive occasions the despooler finds no work when it polls a queue, it changes to the slow rate. The number of times it should try before changing to the slow rate is set by the -DECAY option. When the despooler finds work in the queue, it changes back to the fast poll rate. If a spool queue is unavailable when the despooler tries to search it (a disk could be offline for example), it tries again after the interval defined by the -QUEUE_SEARCH_INTERVAL option.

▶ $\left\{\begin{array}{l}\text{DISK_SCAN}\\ \text{DSCN}\end{array}\right\}$n

Sets the interval at which the despooler takes account of configuration changes by reconstructing its list of disks on remote nodes. This command affects only pre-Rev. 21.0

format spool queues that are configured by node name, and is therefore used only in conjunction with the command NODE -PRE_REV_21. *n* is the number of minutes. The default is 10.

▶ NODE [nodename] [options]

Specifies the node on which a spool queue is located. You can have none, one, or several NODE commands in an environment file provided that the combined total of NODE and DISK commands does not exceed 100. *nodename* is a PRIMENET node name. If you omit *nodename,* the default is the local node. If there are no NODE or DISK commands in the environment file, the default is NODE -LOCAL.

The options of the NODE command are as follows:

Option	*Description*
{ -LOCAL -LCL }	An alternative way of specifying the local system. It does not require the node name, and is therefore useful if the system is not in a network.
{ -PRE_REV_21 -PRE }	Indicates that the specified node is Rev. 20.2 or earlier. Spool queues at Rev. 21.0 are identified by node name. Spool queues at Rev. 20.2 and earlier are identified either by node name or by disk partition name. If you use this option, *all* the disks on the remote node, that are added to the local node, are included in the table of disks that the despooler searches for spool queues. The table is updated at regular intervals as specified in the DISK_SCAN command. If you do not use this option, any node that is identified by node name is initially assumed by the despooler to be a Rev. 21.0 node. When the despooler phantom finds that the node is not a Rev. 21.0 node it automatically adds all known disks on that node to its table of pre-Rev. 21.0 queue tables.
{ -RING -RNG }	Tells the despooler to look for a spool queue on all nodes, except the local node, that are in the PRIMENET ring at the time the despooler is started. This option enables you to configure the despooler so that it looks for spool queues in all possible places on the ring. Although this simplifies administration, it can lead to degraded performance through looking for queues that do not exist.

The remaining options enable you to control how frequently the despooler checks the spool queues for work. These are the same options as in the POLL_RATE command. The options are as follows:

Option *Description*

{-FAST_INTERVAL} n
-FI

Fast poll rate. *n* is the interval in minutes. The default is 1 minute.

{-SLOW_INTERVAL} n
-SI

Slow poll rate. *n* is the interval in minutes. The default value is 5 minutes.

{-QUEUE_SEARCH_INTERVAL} n
-QSI

Length of time the despooler phantom is to wait before it retries an unavailable spool queue. *n* is the length of time in minutes. The default is 10 minutes.

{-DECAY} n
-DCY

Number of times the despooler phantom is to poll a spool queue unsuccessfully at the fast rate before it changes to the slow rate. The default is 10 attempts.

The despooler uses the fast poll rate as long as it finds work each time. If on successive occasions the despooler finds no work when it polls a queue, it changes to the slow rate. The number of times it should try before changing to the slow rate is set by the -DECAY option. When the despooler finds work in the queue it changes back to the fast poll rate. If a spool queue is unavailable when the despooler tries to search it (a disk could be offline for example), it tries again after the interval defined by the -QUEUE_SEARCH_INTERVAL option.

▶ {POLL_RATE} [options]
PLRAT

This command sets the poll rate parameters that determine how frequently the despooler polls each spool queue to find work. You can give different values in individual NODE and DISK commands.

The options to the POLL_RATE command are as follows:

Option *Description*

{-FAST_INTERVAL} n
-FI

Fast poll rate. *n* is the interval in minutes. The default is 1 minute.

{-SLOW_INTERVAL} n
-SI

Slow poll rate. *n* is the interval in minutes. The default value is 5 minutes.

{-QUEUE_SEARCH_INTERVAL} n
-QSI

Length of time the despooler phantom is to wait before it retries an unavailable spool queue. *n* is the length of time in minutes. The default is 10 minutes.

$\left\{\begin{matrix}\text{-DECAY}\\\text{-DCY}\end{matrix}\right\}$ n

Number of times the despooler phantom is to poll a spool queue unsuccessfully at the fast rate before it changes to the slow rate. The default is 10 attempts.

The despooler uses the fast poll rate as long as it finds work each time. If on successive occasions the despooler finds no work when it polls a queue, it changes to the slow rate. The number of times it should try before changing to the slow rate is set by the -DECAY option. When the despooler finds work in the queue it changes back to the fast poll rate. If a spool queue is unavailable when the despooler tries to search it (a disk could be offline, for example), it tries again after the interval defined by the -QUEUE_SEARCH_INTERVAL option.

▶ $\left\{\begin{matrix}\text{PRIORITY\_TO\_LOCAL\_QUEUE}\\\text{PLQ}\end{matrix}\right\}$

Gives priority to jobs in the local queue. Normally, when a despooler finishes printing a file, it checks the same queue to see if it contains more spool requests that it can service, and checks queues for jobs in strict rotation. Thus a long print queue on a remote machine could block despooling from the local queue.

The -PRIORITY_TO_LOCAL_QUEUE option forces the despooler to search the local queue whenever it finishes printing a file, provided that the next poll of the local queue is in fact due at that time.

▶ $\left\{\begin{matrix}\text{SWO\_TIMES\_TO\_CHECK}\\\text{SWOTTC}\end{matrix}\right\}$ n

Defines the maximum number of times the despooler should be instructed to sleep at each end of file (EOF), during the Spool_While_Open function. The range for *n* is 0 through 32767 inclusive, with 1 the default. A value zero causes no special EOF processing, and the despooler stops printing as soon as EOF is encountered. With *n* set to 1, the despooler sleeps once on EOF, and if EOF occurs immediately the file is re-read, stops printing. With *n* set to 2, the despooler sleeps and retries twice on EOF, and if each retry has an error EOF, the despooler stops printing.

▶ $\left\{\begin{matrix}\text{SWO\_WAIT\_TIME}\\\text{SWOWT}\end{matrix}\right\}$ n

Defines the time the despooler waits when it reaches the end of file (EOF) during a Spool While Open function. *n* is the period the despooler is to sleep in seconds, and is in the range 1 through 600 seconds, with 5 seconds the default.

Formatting the Printer Output

The commands in this group specify how the output is formatted when a file is printed.

▶ $\left\{\begin{matrix}\text{AUXILIARY}\\\text{AUX}\end{matrix}\right\}$ **-element n string**

Enables you to pass environment parameters to user written print handlers, the header page, and other user replaceable routines. The environment parameters are read from the environment file, and are inserted into an auxiliary structure in external common memory, from where they can be accessed by the users own routines. Note, the auxiliary structure is not used by the standard despooler, but exists solely for the benefit of user self-written routines. The content and use of these parameters are the user's responsibility.

By using the AUXILIARY command in a printer environment definition, you can specify the element in the auxiliary structure that is to hold a defined string.

In the above statement format, *n* has a range 1 through 5, and is the element in the auxiliary structure, holding the string. The string contents has a maximum of 160 characters. The auxiliary structure, held in SPOOL* > SOURCE > AUXILIARY.INS.PL1, contains an array of up to five elements with their string contents, stored as Char(160) var strings. The declaration of the structure is:

```
dcl auxiliary(5) char(160) var static external;
```

▶ **DEVICE handler [options]**

Specifies the print handler for the despooler phantom to use. *handler* is the name of the print handler routine. You must specify a device, except when you have specified a plotter with PLOT in which case you must not give the DEVICE command.

The following standard print handlers are supplied by Prime:

DATAPRODUCTS
DIABLO
DPTX
NEC
NEC_OAS
POSTSCRIPT
PRINTRONIX
QMS
TTY
QUME
QUME_OAS

You can add print handlers for other devices, as explained in Appendix B, Writing a Print Handler. The code for the print handlers is located in the file SPOOL* > DESPOOL_LIBRARY.RUN, which is a library class EPF.

Note

> For a PRINTRONIX plotter you must give a DEVICE command, and may need to give an ASYNC command. You do not need to give either the DEVICE or ASYNC commands for a Versatec or Gould plotter. Give the special PLOT environment command instead. For details, see the description of PLOT later in this chapter.

You can specify information that the despooler phantom needs to initialize the output device, and other output device information, in the options to the DEVICE command.

The options to the DEVICE command are shown on the next page.

Option — *Description*

-EVFU pathname

Specifies the pathname of a file that contains an Electronic Vertical Format Unit definition. If -EVFU is given with no *pathname*, a standard EVFU format is used, with channel 1 assigned to line 1 and no other channels defined. -EVFU_OFF Indicates that the printer has no available EVFU. In this case the software emulates the EVFU function. This is the default.

-EVFU_OFF

-EVFU_OFF Indicates that the printer has no available EVFU. In this case the software emulates the EVFU function. This is the default.

-DICT pathname

Specifies a file that contains a list of pathnames of files that in turn contain PostScript procedures. (There can be several such files, so that the system can support several PostScript printers). The PostScript procedures are downloaded to the printer when the despooler phantom is started. At the start of each print job, the printer is checked to make sure that the procedures are still loaded. (The printer could have been switched off and back on again, for example.) If it is found that the procedures are not present, they are reloaded.

The default *pathname* is SPOOL*>POSTSCRIPT_INIT. Users can add the pathnames of additional PostScript procedure files to the SPOOL*>POSTSCRIPT_INIT file.

{ -NO_LINEFEED
-NFL }

Indicates that the device generates a line feed when it does a carriage return, and that the despooler therefore does not need to insert line feed characters. Note that a printer that requires this option cannot perform overstriking.

-PI

You must specify a PRINTRONIX device to use this option. It indicates that jumper W5 on logic board A is made, that is, that the paper instruction is enabled. See the Printronix printer manual for details.

{ -STATIONERY_FEED_TYPE } { MANUAL
-SFT CUTSHEET
CONTINUOUS }

Specifies the type of paper feed mechanism. The default is CONTINUOUS.

{ -UPCASE
-UC }

Indicates that the output device requires the data stream to be converted to uppercase.

-KSR	Indicates that the device has an attached keyboard.

▶ DPTX printer_name

DPTX (Distributed Processing Terminal Executive) specifies a suitable IBM *printer_name* when the DEVICE command has previously been used to select the DPTX print handler. Printer_name must already be defined in the DPTX configuration.

▶ FORMAT [options]

Sets up all aspects of the page format that is used when files are printed.

The options to the FORMAT command are as follows:

Option	*Description*
-BOTTOM_MARGIN **-BM** n	Specifies number of blank lines to be left at the foot of each page. *n* is the number of lines. The default is 2 lines.
-LEFT_MARGIN **-LM** n	Specifies width of left margin. *n* is the number of characters. The default is 0.
-LENGTH **-L** n	Sets number of lines per page (form length).Your choice depends on the type of stationery used and any special requirements. *n* is the number of lines. The default is 66.
-PAD_CHAR **-PCH** n	Specifies the character used to fill the last -PCH byte of an odd length line that is sent to a device with a parallel interface. The correct type of ASCII code for the printer must be used. *n* is the ASCII code for the chosen character. Default is a space character (ASCII '240). The appropriate type of ASCII for the printer must be used. This is usually ASCII-8.
-RIGHT_MARGIN **-RM** n	Specifies width of right margin as number of characters. The value of the default is 0.
-TOP_MARGIN **-TM** n	Specifies number of blank lines to be left at the head of each page. *n* is the number of lines. The default is 4.
-WIDTH **-W** n	Specifies page width. *n* is the number of characters. The default is 132.

▶ HEADER [n] [options]
HDR

Each file that is printed can have a number of additional header pages. The HEADER command sets up the header characteristics. *n* is the number of header pages required. Permitted values of n are 0, 1, 2. The default is 1. The options to the HEADER command are shown on the next page.

Option — *Description*

-FACING_BANNER
-FB

Ensures that the banner page always faces the same way on fan folded listings. It works by performing a page eject between jobs if necessary.

-SUPPRESS_FILE_INFO
-SFI

Prevents a file pathname, and time and date the file was modified, from being printed on the header and trailer pages.

In addition you can use any one, but not more than one, of the following options, to specify the printer characters used to form the large characters in the header. If none of the following options is used the default is -CHAR W.

Option — *Description*

-CHAR c

Identifies a specific character *c* to be used. The default is W.

-ROLLING_BANNER_CHARACTER
-RBC

Uses a different character each time a file is printed to equalize wear on the printer.

-SAME

Produces each large letter from the corresponding character. That is, builds large As in the banner from the A character, large Bs from the B character and so on.

▶ PLOT

Specifies that the output device is a Versatec or Gould plotter. A special PRIMOS routine controls this type of plotter, so there is no need to specify line protocol and configuration; specifying a print handler is optional.

▶ MAP mapping filename { -DEFAULT | -ALTERNATE }

Specifies the name of the file that holds the available mapping. *mapping* corresponds to the string you called in the SPOOL command -XLATE option, and is a maximum of 32 characters. The first character must be alphabetic and the remainder alphanumeric, including, . $ or _. *filename* is where the definition of the map is filed in SPOOL*>MAPPINGS.

An example follows:

```
MAP IBM ECS_TO_EBCDIC_MAP
```

where IBM is the map called with the SPOOL command, for example,

```
SPOOL TEMP.FILE -XLATE IBM
```

ECS_TO_EBCDIC_MAP is the filename in which the IBM map definition is filed in SPOOL*>MAPPINGS.

If you give -DEFAULT, the statement defines the default mapping for the environment. If no statement has -DEFAULT, the *first* MAP statement in the environment file, excluding any marked -ALTERNATE, defines the default. The default map can be selected explicitly by name.

If you give -ALTERNATE, the statement defines the mapping selected when you spool print files using the -XLATE option without *mapping*. If no MAP statement has an -ALTERNATE option, then the first MAP statement which is NOT the default is the alternate mapping.

There can be a maximum of 64 MAPs in a single environment file. See Chapter 2, Setting Up the Spooler Subsystem for a discussion on Character Mapping Files.

If the SPOOL command is given without the -XLATE option, character mapping is attempted in the following order:

1. Character mapping is performed by the map which has been defined as the default in the environment file.
2. If no default map has been defined, then STANDARD_MAP is used, if it exists.
3. If neither 1 or 2 exists the despooler emulates the Rev. 21.0 version. That is, the file is read assuming:
   - Space compression has been used.
   - A line of text terminates with a line feed.
   - For each form feed, in the file, a page throw is to be generated.
   - The input files are in ECS and the output is in ASCII.

▶ $\left\{\begin{array}{l}\textbf{MESSAGE}\\ \textbf{MES}\end{array}\right\}$ **text**

Prints a message on each header page. *text* is the message that is to be printed. It can be a maximum of 160 characters.

▶ **SET_FONT -NAME font_name -SEQ escape_sequence** $\left\{\begin{array}{l}\textbf{-STRING}\\ \textbf{-REP}\end{array}\right\}$ **[-DEFAULT]**

Specifies the font type the printer uses to print a document. *font_name* is a character string used to select the font. *font_name* has a maximum of 32 characters. *escape_sequence* activates the font, and can be found in the printer's instruction book. -STRING defines this escape sequence to be the actual string itself; alternatively, -REP defines the string to be the DENARY (base 10) representation of the characters. -DEFAULT specifies the font type the printer uses when you spool a document without -SET_FONT instructions.

▶ **SET_LANDSCAPE [-DEFAULT]**

Specifies that a printer is capable of printing in landscape. For a given size of paper a printer is printing landscape if text is printed across the longest width. With -DEFAULT specified this command ensures the printer prints in landscape if no paper orientation instruction is given when you spool a document.

```
▶  SET_PAPER_BIN  -BIN ⎧n       ⎫ [-DEFAULT]
                       ⎩MANUAL  ⎭
```

Defines which paper bin the printer takes its paper from. Where *n* is a number 1 through 9. When MANUAL is given, the paper can be fed to the printer manually. -DEFAULT specifies which paper bin the printer selects if you give no argument with the -SET_PAPER_BIN instruction when spooling a document.

```
▶  SET_PORTRAIT  [-DEFAULT]
```

Specifies that a printer is capable of printing in portrait. For a given size of paper a printer is printing in portrait if the text is printed across the shortest width. With -DEFAULT specified this command ensures the printer prints in portrait if no paper orientation instruction is given when you spool a document. This manual is printed in Portrait orientation.

```
▶ ⎧ TRAILER⎫⎧-ON  ⎫ ⎧-PERF n  [ -PERF_CHAR a ]  ⎫
  ⎩ TRLR   ⎭⎩-OFF ⎭ ⎩-NO_PERF                   ⎭
```

Specifies whether or not a trailer page is printed. The default is not to print a trailer. If you choose to print a trailer page the following information is printed:

- Date and time the job was completed
- Request number of the job
- Number of pages printed, excluding headers and footers
- User ID
- Project ID
- Node name
- Pathname of the spooled file

The pathname is not printed if the environment file contains a HEADER -SFI command, or if the SPOOL command uses the -SFI option.

Option	*Description*
-ON	Adds a trailer page to each document.
-OFF	No trailer pages are printed. This is the default.
-PERF n	A number of full lines are printed to mark the perforation between jobs. One line is printed on the perforation itself, and *n* lines are printed both before and after the perforation. The default for n is 2.
-NO_PERF	No perforation marker is printed. This is the default.

-PERF_CHAR a Specifies the character to print to mark perforations. The default is the current rolling banner character.

▶ **WARNING** $\left\{\begin{matrix}\textbf{-ON} \\ \textbf{-OFF}\end{matrix}\right\}$

Specifies whether large print warning messages should be output on the printer to give notification of major events (Starting, Stopping, Dropped, and so on). The default is -ON.

Configuring the Data Line to the Printer

The commands in this group give details of the type of output device, communication protocols, and the interface hardware. The device commands must be consistent with the format commands. Two separately chargeable versions of the Despooler are available that allow you to send information to printers over a Public Data Network (PDN) and across a TCP/IP network. See Appendix D, The PDN Support Option for details and Appendix E, Despooling over TCP/IP.

▶ **ASYNC -LINE n [options]**

Specifies the configuration of the asynchronous serial line that connects to a printer. *n* is the number of the line to which the printer is connected.

Usually each printer in an installation uses a dedicated line that is configured correctly for that printer, during PRIMOS cold start, with the PRIMOS SET_ASYNC command. You need only use ASYNC with printers that use assignable lines.

The ASYNC command is like the PRIMOS SET_ASYNC command, but with the following differences:

- Some options of the PRIMOS SET_ASYNC command can only be used by a System Administrator logged in as SYSTEM, and therefore are not available in ASYNC.
- Many of the options in SET_ASYNC are used to set up lines to VDU terminals and therefore are not relevant for most printers.
- One of the ASYNC options, -CARRIER_DETECT, has no parallel in the PRIMOS SET_ASYNC command.

The ASYNC environment command is useful when you frequently need to move printers, or when you often change the printer attached to one line, as would happen if you had to test different types of printer, and had only one line available. In such cases it could be easier to configure the asynchronous line with ASYNC commands in environment files than with the PRIMOS SET_ASYNC command.

You can use the options of the ASYNC command to control features of the data line itself. For example you can use options to control transmission speed, number of stop bits, and type of parity, or of how the attached device is handled, for example, whether communication is duplex or half-duplex mode.

When you give the ASYNC command the line settings take the default values listed in Table 3-1. You can override these defaults with the specific options.

TABLE 3-1. Default Settings Produced by ASYNC

Default Option	*Explanation*
-CHAR_LENGTH 8	In each byte transmitted, all 8 bits are used to represent a character.
-DATA_SET_CONTROL	Data Set control attribute is set.
-ECHO	Line is in full-duplex mode.
-NO_DATA_SENSE_ENABLE	Data Set Sense (DSS) protocol is disabled.
-NO_ERROR_DETECTION	There is no error detection. The NAK character is not sent back down the line if an input parity error occurs, or input buffer overflow is sensed.
-NO_LINE_FEED	The line does not send a LINEFEED character in response to receiving a RETURN character.
-NO_REVERSE_XOFF	Reverse Flow control (XON/XOFF) is disabled.
-PARITY NONE	Parity is not generated.
-PROTOCOL TRAN	The line uses transparent protocol.
-SPEED 1200	Data transmission rate is 1200 baud.
-STOP_BITS 1	A single stop bit is used.
-XOFF	Sending XOFF (CONTROL-S) from the device to the CPU stops the flow of data to the device. Sending XON (CONTROL-Q) from the device to the CPU starts the flow of data to the device.

Full details of all the options to the ASYNC command are as follows:

Option — *Description*

$\left\{\begin{matrix}\textbf{-CARRIER\_DETECT}\\ \textbf{-CD}\end{matrix}\right\}$ **n**

Makes the despooler phantom check the carrier detect line at regular intervals. If the line is low (for example, if the printer is turned off) the despooler phantom waits.

n is the period between checks, given in milliseconds. The default is 200 milliseconds. On locally connected lines the carrier state reported by PRIMOS corresponds to the DTR line.

This option has no parallel option in the PRIMOS SET_ASYNC command.

$\left\{\begin{matrix}\textbf{-CHAR\_LENGTH}\\ \textbf{-CL}\end{matrix}\right\}$ **n**

Sets the number of bits that represent a character in each byte of data transmitted. *n* can be 5, 6, 7, or 8. The default is 8.

-DATA_SENSE_ENABLE
-DSE

Enables Data Set Sense (DSS) protocol. Sometimes referred to as reverse channel protocol.
Default is -NO_DATA_SENSE_ENABLE.

-DATA_SET_CONTROL
-DSC

Sets Data Set Control (DSC). DSC is a handshaking signal -DSC that is required by modems and port selectors. This is the default.

-DATA_SET_SENSE **HIGH**
-DSS **LOW**

Sets the logic level recognized by the DSS protocol. The DSS protocol must be enabled with the -DATA_SENSE_ENABLE option.

-ECHO

Sets the line to full-duplex.

-ERROR_DETECTION
-ERRDET

Enables error detection. A NAK character is sent if a parity error, or input buffer overflow, is sensed.
Default is -NO_ERROR_DETECTION.

-LINE_FEED
-LF

The line sends a LINEFEED character in response to receiving a RETURN character. This is useful for VDUs but not usually needed by printers. The default is -NO_LINE_FEED.

-NO_DATA_SENSE_ENABLE
-NODSE

Disables the Data Set Sense (DSS) protocol. This is the default.

-NO_DATA_SET_CONTROL
-NDOSE

Disables Data Set Control (DSC). The default is -DATA_SET CONTROL.

-NO_ECHO

Sets the line to half-duplex. This is needed for PostScript printers.

-NO_ERROR_DETECTION
-NOERRDET

NAK character is not sent if there is a parity error or input buffer overflow. -NOERRDET is the default.

-NO_LINE_FEED
-NOLF

The line does not send a LINEFEED character in response to receiving a RETURN character. -NO_LINE_FEED is the default.

-NO_REVERSE_XOFF
-NOREVXOFF

Disables Reverse Flow control. -NO_REVERSE_XOFF is the default. See -REVERSE_XOFF for details of Reverse Flow control.

-NO_XOFF

Inhibits XOFF (CONTROL-S) and XON (CONTROL-Q) to prevent start and stop of data flow from the computer to the device.

$$\left\{\begin{array}{l}\textbf{-PARITY}\\\textbf{-PAR}\end{array}\right\}\left\{\begin{array}{l}\textbf{ODD}\\\textbf{EVEN}\\\textbf{NONE}\end{array}\right\}$$

Defines the type of parity to be used, or disables parity generation. Default is NONE

$$\left\{\begin{array}{l}\textbf{-PROTOCOL}\\\textbf{-PRO}\end{array}\right\}\textbf{[name]}$$

Sets the line protocol. Permitted protocol names are as follows:

TTY Standard terminal protocol. Typically this is used for serial printers, such as QUME.

TRAN Transparent protocol. This is the default, and is used if you specify -PROTOCOL without an explicit protocol name. TRAN is used when it is not necessary to echo input, convert carriage returns to line feeds, or specifically acknowledge carriage returns and line feeds. All characters pass as data unless XON/XOFF flow control is set.

TTY8 Identical to TTY except that ASCII-8 is supported. TTY8 adapts the standard terminal protocol, TTY, for devices that use Prime ECS. Typically this protocol is used for PostScript printers.

TTY8HS As TTY8, but for older model AMLC boards (5052 or 5054).

TTYUPC As TTY but data is transmitted in uppercase only. This protocol name avoids sending lowercase output to devices that cannot print lowercase characters.

TT8BIT Input protocol is as for TTY, but output supports full 8-bit ASCII terminals. All control characters are handled in the same manner as the TTY protocol.

TTYHS As TTY but permits more frequent interrupts. This is not used with the 5154 AMLC board.

TTYHUP As TTYUPC but permits more frequent interrupts. This is not used with the 5154 AMLC board.

TRANHS As TRAN but permits more frequent interrupts. This is not used with the 5154 AMLC board.

Further information on protocols can be found in the System Administrator's Guide,

-REVERSE_XOFF
-REVXOFF

Enables Reverse Flow control for a line connected to an ICS controller. If the PRIMOS input queue is full, the ICS controller can send an XOFF character to the device to prevent more data from being sent to the computer. When the input queue has space available the ICS controller can send an XON character to resume transmission. This option is not usually used with printers. The default is -NO_REVERSE_XOFF.

-SPEED [value]

Sets the speed of the line. Value can be any of the supported set_async speeds, plus J1, J2, J3 and clock.

Numeric values set the baud rate directly. The default is 1200. CLOCK uses the programmable clock, as specified by the PRIMOS AMLCLK configuration directive, this is usually 9600. J1, J2 and J3 use the jumper speeds.

$\left\{\begin{array}{l}\textbf{-STOP\_BITS}\\ \textbf{-SB}\end{array}\right\} \quad \left\{\begin{array}{l}1\\ 2\end{array}\right\}$

Specifies whether to use one or two stop bits.

-XOFF

Enables XOFF (CONTROL-S) and XON (CONTROL-Q) to allow control of data flow from the computer to the device.

▶ $\left\{\begin{array}{l}\textbf{PARALLEL}\\ \textbf{PLL}\end{array}\right\}$ **PRn [-CENTRONICS]**

Specifies a parallel interface. PR*n* identifies the printer. *n* is the line number on the parallel interface and is in the range 0 to 3 inclusive. The -CENTRONICS option informs the despooler phantom of the different canble wiring used by the Centronics interface.

▶ PLOT

Specifies either, that the output device is a Versatec, or Gould plotter. A special PRIMOS routine controls the Versatec, or Gould plotters, so there is no need to specify line protocol and configuration, specifying a print handler is optional.

The -PLOT option should not be used in conjunction with the -NO_FORMAT option. They are both types of print formats.

▶ TTY [n]

Allows you to direct the despooler phantom to send the contents of a file to a terminal buffer, or to a Centronics printer.

If you omit *n*, the file is sent to a terminal buffer. Usually the despooler runs as a phantom process, possibly with the log file open (as a COMO file), in which case output is directed to the log file. This option is useful as a debugging aid, when you are developing a print handler. You can start a despooler and see its output on your terminal.

If you include *n*, it indicates that output is to go to a Centronics printer. *n* can take either of the values 1 or 2, which refer to the first Centronics printer and the second Centronics printer, on the interface board.

Logging Spooler Activity

When the PROP command starts a despooler phantom, it reads and validates the environment file, and writes a copy of the environment file with the following pathname:

```
SPOOL*>LOG>environment.LOG
```

where *environment* is the environment name.

▶ LOG [option]

Determines the further logging action taken once the despooler is successfully started. The options follow:

Option	*Description*
-ON	The file is not closed and records information about the despooler's activities. A single line of accounting information is recorded for each job. The next startup of the same environment overwrites the file.
-OFF	The file is closed after startup. The next startup of the same environment overwrites the file.
-SAVE	Same as -ON except that the date and time are appended to the name of the log file, as follows: SPOOL*>LOG>environment.LOGdatestamp where *environment* is the environment name and *datestamp* is the full date in the format yymmddhhmmss. This file is not overwritten the next time the environment is started, because a different date stamp is generated for each successive filename. You need to delete old log files occasionally, otherwise the disk becomes full.

Displaying Spooler Status

By using the SPOOL -LIST -DETAIL or SPOOL -LIST -FULL commands the percentage spooled values of files are displayed. See Appendix A, Viewing the Spool Queue, for further details. Each time a new page is started in the despooler, the number of records processed so far is compared against two values. If the number of records processed since the last report is less than the reporting increment defined in the printer environment file, the percentage spooled value is not updated. If the number of records processed is greater than the reporting increment, then the number processed so far is compared against the file size, and the percentage spooled is calculated and displayed. The same check is done for multiple copies, as each copy is started.

▶ $\left\{\begin{array}{l}\textbf{PERCENT\_SPOOLED\_INCREMENT}\\ \textbf{PSI}\end{array}\right\}$ **n**

Determines the reporting increment used in the calculation of percentage spooled. The value *n* is the number of records that must be processed before the percentage spooled value is updated; the default is 10 records. If you do not wish to see the percentage spooled values you can set a large value for *n*, for example 32767, which represents 10 or more boxes of paper for a single file.

Modifying Printer Environment Files

The system administrator should create environment files for each combination of printer, paper type, page format, location, and so on. It should rarely be necessary to edit an environment file once all the necessary environment files are created and tested. You can stop an active environment and start a new one in its place with the PROP -RESET command.

If you need to change an environment file, use a text editor, such as ED or EMACS. In case you make mistakes during your editing be sure to make a backup copy of the original file. For example, copy the file to be modified, and save it under a different name. Note that the altered environment does not come into effect until the next time the environment is started.

4

ROUTINE OPERATIONAL PROCEDURES

Introduction

This chapter explains how to use the SPOOL and PROP commands to carry out many of the tasks that are often required of an operator. For a summary of the PROP and SPOOL commands and all the available options, see Appendix A, Summary of Spool and Prop Commands. The SPOOL command is also explained in the *PRIMOS Commands Reference Guide,* and the PROP command is explained in the *Operator's Guide to System Commands.*

The operations described in this chapter are as follows:

- Monitoring the status of despooler phantoms
- Controlling despooler phantoms
- Changing to another printer environment for a particular printer
- Modifying a printer environment
- Listing the files in the spool queue
- Canceling the printing of a file
- Modifying spool requests
- Hastening the printing of a file

Monitoring the Despooler Status

To find out the status of each of the available printer environments type PROP -STATUS -ALL. In response, a list of environment names is displayed.

The current status is displayed beside each environment name, for example

```
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
PR1          Stopping
DOC          Hanging
QMS          Idle
LQP          Printing(LETTER.RUNO: page 1, copy 1 of 2
               Request: 87 from node: SYSA)
CEN          Not Started
```

For each despooler that is listed, the status can be any of those in the following list:

Status	*Meaning*
Aborting	Stops printing, but leaves the request in the print queue.
Backing Up	Backs up to a suitable point to restart a print job.
Dropping	Stops printing, removes the request from the queue.
Hanging	Despooler is suspended and waiting for the command to continue.
Idle	Despooler phantom has been started but at the moment there are no suitable requests in the spool queues for it to print.
Lineup	Despooler is suspended following a paper alignment check.
Not Started	There is an environment file for the despooler phantom, but the environment has not been started, or has been started previously but has since been stopped.
Printing	A file is being printed on the printer that is specified in the corresponding environment file.
Reset	Despooler is stopping, and will start up with a new environment.
Restarting	Environment is starting after being suspended.
SWO	Spooling while open <filename>, Page <p>, Copy <a> of <b>. A file that has been queued using the -SPOOL_WHILE_OPEN option is being printed.
Sleeping	A file that has been queued using the -SPOOL_WHILE_OPEN option is in the process of printing, but the despooler is waiting for more to be written to the end of the file.
Starting Up	Environment is starting.
Stopping	Environment is stopping.

To see a list of all active environments, excluding those that are stopped, or have not been started, give the command PROP -STATUS.

The files that contain environment information are stored in the SPOOL* directory. To view an environment issue the command PROP environment -DISPLAY. `PROP environment -DISPLAY,`

where *environment* is the environment name. In response, the status of the environment is displayed and the contents of the environment file are displayed. You can also view environment files with either the SLIST command, or with a text editor such as ED or EMACS, but then you must give full pathnames, unless you are attached to the SPOOL* directory.

The pathname of each environment file is:

```
SPOOL*>environment.ENV
```

where *environment* is an environment name, such as DOC or LQP.

See Chapter 3, Defining Printer Environments, for a description of the contents of an environment file and an explanation of how environment files are used.

If an environment file has been modified since the despooler was started then the information in it does not describe the environment that is active. When a despooler is started a file is logged with the environment data that is used. The exact name of the file depends on the options used with the LOG command in the environment file. For details see Chapter 3, Defining Printer Environments. The log file contains definitive information about the active despooler.

Controlling a Despooler Phantom

This section explains how to use the PROP command to perform the following tasks:

- Start a despooler phantom.
- Suspend a despooler phantom.
- Make a suspended despooler phantom continue.
- Stop a despooler phantom.

How to Start a Despooler Phantom

To start up a phantom to control an environment, issue the command:

```
PROP environment -START.
```

You should use this command to start a phantom initially, as well as to restart a phantom that has been stopped.

When, as is usually the case, you start a despooler phantom from the supervisor terminal, it takes its name from the environment file. For example if you start the LQP environment from the supervisor terminal, and the environment file is named LQP.ENV, the despooler phantom is named LQP. If you start a despooler phantom from any other terminal, your user-ID becomes the phantom's name.

In the following example, user FRED, who is logged in at a user terminal, starts the despooler phantom that controls the printer environment LQP.

```
OK, PROP LQP -START
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
*** FRED (User 102 on SYSA) at 09:29
Despooler for LQP ready
```

How to Suspend a Despooler Phantom

Use the -HANG option of the PROP command to suspend temporarily a phantom that is controlling a particular printer environment.

Ordinarily you need to suspend a phantom to attend to any mechanical problem that arises on a printer. Also, to perform some routine tasks, such as aligning forms, the printer must be online but not printing.

The command that suspends a phantom is as follows:

```
PROP environment  -HANG  [-FINISH]
                         [-NOW   ]
                         [-IDLE  ]
```

The -FINISH option suspends the phantom when it has printed the current file. The -NOW option suspends the phantom immediately. The print request remains in the spool queue and the file is printed when you restart the phantom. The default is -FINISH.

The -IDLE option suspends the phantom when there is nothing left for it to print.

In the following example, the phantom that controls the environment LQP is told to suspend activity after printing the current file.

```
OK, PROP LQP -HANG -FINISH
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
*** FRED (User 102 on SYSA) at 09:29
Hanging
```

Note that the name of the phantom appears in the message. Only if the despooler phantom was started at the supervisor terminal, or by a privileged process, is the phantom name the same as the environment name.

How to Restart a Suspended Despooler Phantom

To cause a hung environment to continue printing, issue the command:

```
PROP environment -CONTINUE
```

The despooler phantom continues from wherever it left off, whether it was printing a file or scanning queues. In the following example, the phantom that controls environment LQP is told to continue.

```
OK, PROP LQP -CONTINUE
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
```

The -CONTINUE option does not produce a specific message saying that the phantom is continuing. This is unlike the -START, -HANG, and -STOP options.

How to Stop a Despooler Phantom

The command that stops a phantom is as follows:

```
PROP environment  -STOP  [ -FINISH ]
                         [ -NOW    ]
                         [ -IDLE   ]
```

The -FINISH option stops the phantom when it has printed the current file. -FINISH is the default.

The -NOW option stops the phantom immediately. The print request remains in the spool queue and the file is printed when you start the phantom again.

The -IDLE option stops the phantom when there is nothing left for it to print.

In the following example, the phantom that controls the environment LQP is told, by default, to stop after printing the current file.

```
OK, PROP LQP -STOP
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
*** FRED (user 102 on SYSA) at 10:33
Environment for LQP stopping

Phantom 102: Normal logout at 10:33
Time used: 01h 03m connect, 01m 23s CPU, 00m 48s I/O
```

Changing to Another Printer Environment

Starting a new environment for a printer involves two processes. The currently active phantom must be stopped, and the new phantom must be started.

If you need to change paper or to make physical adjustments to the printer, you should stop one phantom and start another with separate commands, and make whatever changes are needed during the time between the stop and the start instructions.

In the example that follows, the phantom controlling environment LQP is stopped and a new phantom that controls environment DOC is started.

```
OK, PROP LQP -STOP -FINISH
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
*** FRED (user 102 on SYSA) at 09:40
```

```
Environment for LQP  stopping

Phantom 102: Normal logout at 09:40
Time used: 00h 10m connect, 00m 21s CPU, 00m 07s I/O

OK, PROP DOC -START
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
*** FRED (user 103 on SYSA) at 09:41
Despooler for DOC ready
```

There may be a brief delay between the time you give a command and its execution.

You can use PROP -STATUS to make sure that the phantom has started.

If you do not need to make any changes to the printer you can stop one phantom and start another with a single command, (PROP env1 -RESET env2), as shown in the following example:

```
OK, PROP LQP -RESET DOC
[PROP Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
OK,
*** FRED (user 102 on SYSA) at 09:40
Environment for LQP  stopping

Phantom 102: Normal logout at 09:40
Time used: 00h 10m connect, 00m 21s CPU, 00m 07s I/O

*** FRED (user 103 on SYSA) at 09:40
Despooler for DOC ready
```

Modifying a Printer Environment

You should rarely need to change environment files. It is best to have separate environments defined for all the types of printing that are needed.

To modify an environment, you must edit the environment file using an editor such as ED or EMACS. Environment files are described in Chapter 3, Defining Printer Environments.

For example, to modify the active environment named WHITE, first edit the environment file. In this case the filename is SPOOL*>WHITE.ENV

Make sure that you save the altered version of the file, and check that it is correct by issuing the command `PROP WHITE -VERIFY`.

If the file is correct you can replace the active version of the WHITE environment with the new version by issuing the command `PROP WHITE -RESET`.

The updated environment is sent to the despooler.

If the phantom is printing a file, the new environment does not take effect until it finishes printing the file. If the phantom is not printing a file, the new environment takes effect immediately. When the new environment takes effect, a message similar to the following, is displayed on your terminal

```
*** FRED (User 97 on SYSA) at 9:43
Despooler for WHITE ready
```

Listing the Entries in the Spool Queue

To view the entries in the spool queue, issue the SPOOL command with the -LIST option as follows:

```
OK, SPOOL -LIST -DETAIL
[SPOOL Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
System SYSA
Request  Time   User                    Copies  Size  State
-------- -----  ----------------------- ------  ----- ------
74       20:36  JSMITH                  2       2     (Printing)
  File name    <ADMIN>ACCOUNTS>CHECKS
  Attributes   PRINT_ROOM
  Options      -XLATE MATHS
  Despooler    WHITE on node: SYSB

80         20:45 JBROWN                 1       150   (66% Printed)
   File name    <ADMIN>ACCOUNTS>OFFICE
   Attributes   DOC
   Options      -NO_COPY -NOTIFY -XLATE
   Despooler    DOC on node: SYSC

85         20:51 JJONES                 1       200   (Printing)
   File name    <ADMIN>ACCOUNTS>SALES
   Attributes   DOC
   Options      -SPOOL_WHILE_OPEN -NOTIFY
   Despooler    DOC on node: SYSD
```

This example shows that user JSMITH has one request in the spool queue, for the PRINT_ROOM printer, and that the file is currently printing. JSMITH has also called Character Mapping using the translation MATHS on the file. The second user, JBROWN, has one request in the queue, with 66% printed at the DOC printer. The third user, JJONES, has used the -SPOOL_WHILE_OPEN option on one request to enable printing while the file is open for writing to disk.

Note that privileged users can see all the requests in the spool queue, whereas other users can see only their own requests. It is possible for the system administrator to set up the system to allow all users to see the entire spool queue. For details, see Chapter 2, Setting Up The Spooler Subsystem.

For a less detailed summary of the spool queue use the command SPOOL -LIST [-BRIEF].

See Chapter 2, Viewing the Spool Queues for details on how to configure the display from a SPOOL -LIST command.

Canceling the Printing of a File

Privileged users, SYSTEM (the supervisor terminal) and members of .SPOOL_ADMINISTRATORS$, can cancel any pending print request and stop the printing of any file that is being printed on the local system. Other users can cancel only their own print requests.

You may wish to stop a file printing because

- A file that is intended to drive a dot matrix printer has been sent to a line printer. The result is a large amount of meaningless output.
- A large file has just begun to print. You may wish to stop the large file, force a number of small files to be printed, and only then allow the large file to continue printing. You may wish to use the PROP command -SUSPEND and -RELEASE options to do this.

File Not Yet Printing

To remove a request from the spool queue, use the following command:

```
SPOOL [ -ON node_name     ]  -CANCEL  [ req_no [req_no ... ] ]
      [ -DISK_disk_name   ]           [ -ALL                 ]
```

req-no is the number of the file in the print queue. To view the files in the print queue, use the SPOOL -LIST command as explained earlier in this chapter. The -CANCEL -ALL option removes all your own entries from the print queue.

node-name in the -ON option specifies the node on which the spool queue is located.

disk-name in the -DISK option specifies a disk on which a pre-Rev. 21 spool queue is located.

File Has Started Printing

To stop any more of the file from being printed, and at the same time remove the file from the spool queue, you can give the SPOOL -CANCEL command as described in the previous section. Alternatively, give the command PROP environment -DROP.

The advantage in using the PROP command is that you do not have to know the number of the print request in the spool queue.

You may need to stop a file from printing, but leave the request in the spool queue. For example a large file could be printing, when several smaller, but more urgent files are waiting to printed. In that case give the command PROP environment -ABORT.

You can then force other files to be printed before you restart the aborted file. You can do this using one of the methods below.

- Add the -RUSH option to print requests that need to be dealt with first.
- Specify a defer time for the request that you have just aborted. Note that if there is a previously requested suspension (-HANG), or shutdown (-STOP), pending when you use the -ABORT option, the phantom suspends operation, or shuts down, as soon as the file that is printing is aborted.
- Use the PROP command -SUSPEND option to suspend the current printing job, and force other files to be printed. You can then release the suspended job with -RELEASE, and restart it from the top of the page where it was suspended.

Only an operator can issue the -DROP and -ABORT options to the PROP command.

Note

It is not possible to cancel a request that is already printing on a remote node.

Modifying Spool Requests

Users frequently need to modify their print requests after they have issued the SPOOL command. Some possible reasons are to

- Change form type, or the number of copies required.
- Change the time to which printing is deferred.
- Switch destinations (the printer where the file will print).

Occasionally, a user mistypes part of the command, for example spells an attribute incorrectly, and the print request has to be modified before it can be dealt with.

In all cases you should use the -MODIFY option of the SPOOL command. You can issue the -MODIFY option of SPOOL to change any characteristics of your own print requests. However, only privileged users, that is, SYSTEM and members of the .SPOOL_ADMINISTRATOR$ ACL group, can modify other users' requests.

Use the following command format to change the printing characteristics of a file in the spool queue:

```
SPOOL [ -ON node_name   ]   -MODIFY   req_no [ options ]
      [ -DISK partition ]
```

req-no is the number given to the request by the Spooler subsystem. You can find the request number given to your request by looking at the SPOOL -LIST display.

options are any SPOOL options that define job characteristics.

You must use the -ON option or the -DISK option whenever the file you wish to modify is not in the local spool queue.

The -ON or the -DISK option identifies the queue holding the request to be modified. -DISK is used for a pre-Rev. 21.0 queue. You cannot change the spool queue by using -MODIFY, that is move a file from SYSA's queue to SYSB's queue.

Hastening the Printing of a File

The -RUSH option of the SPOOL command tells the despooler phantom to deal with the print request with the -RUSH priority before it starts on the next request in its queue. You must be a privileged user to give the -RUSH option.

The -RUSH option has the following characteristics:

- It does not stop a file that is currently printing. The file with the rush priority prints after the current file finishes.
- If there is more than one file with the rush priority, they are printed in the order that the requests were put in the spool queue, but before other requests.
- Use of the -RUSH option is restricted to the system console and the users in the .SPOOL_ADMINISTRATOR$ ACL group.
- -RUSH does not take priority over queue scanning. Thus, files in other queues may be printed before the file with the rush priority.
- The -RUSH option overrides any -DEFER option.

5

SOLVING OPERATIONAL PROBLEMS

Introduction

This chapter explains how to identify and solve some common software and hardware problems relating to the Spooler subsystem.

The essential first step in solving a problem is to identify it either as a hardware or as a software problem. If a problem arises that is unfamiliar to you, it is often best to leave everything in the error condition until an expert can look into it. If you change anything, you may remove important clues to what is wrong.

Dealing With Files That Will Not Print

Assuming that the printer is switched on and is working correctly, there can be three reasons why a file does not print.

Reason	*Remedy*
Environments are active that consider the request to be eligible, but the despooler phantoms are busy printing other users' files.	Tell the user that the file will print later, when a printer becomes available.
Suitable environments exist but they are not active.	Start a despooler phantom for one of the environments that can print the file. You may need to stop one of the active despooler phantoms to release a printer for the new environment.
No suitable environment is available.	New environments can be created. This is explained in Chapter 3, Defining Printer Environments. If one of the available environments meets the user's needs, change the attributes of the entry in the spool queue with SPOOL -MODIFY, as explained in Chapter 4, Routine Operational Procedures.

Dealing With Mechanical Problems

If there are problems with the operation of a printer, take the following corrective action:

1. Check that the printer is not jammed or out of paper.
2. Check that the printer is ready to print.
3. Check that a suitable despooler phantom is running by using the PROP -STATUS command. If no despooler is controlling the printer, start one by issuing the command PROP environment -START.
4. Make sure that the paper mounted on the printer matches the type called for in the printer's environment. Check the specified paper by typing PROP environment -DISPLAY.

Caution

If more than one parallel printer is connected to a URC controller, and one of them shuts down (because of a blown fuse or power supply problem) then it is possible that the controller will hang. In order to recover you must empty the data buffer in the controller. Usually a warm start recovers the situation, but if that fails, you will need to perform a system cold start. You should repair or replace the faulty printer before the cold start. If that is not possible then you must not start a despooler phantom that controls the faulty printer. The cold start procedure is described in Chapter 2, Setting Up The Spooler Subsystem.

Aligning Forms

A file must be printing to align the paper correctly. You should create a standard line up test file and print it whenever you want to realign the printer. To align the paper to the top of form use the -LINEUP option of the PROP command as follows:

1. Type the command PROP environment -LINEUP.
2. Wait for the despooler phantom to display the following message:

```
Check paper alignment
```

3. Check the alignment of the paper on the printer. The printer should have printed four lines at the top of the page (top of form).
4. If the paper is aligned correctly, skip steps 5 and 6 and proceed to step 7. If the paper is not aligned correctly, go to step 5.
5. Align the paper so that four lines are at top of form.
6. Check the alignment by repeating steps 1 through 4.
7. Issue the command PROP environment -CONTINUE.

Recovering the Printer After a Paper Jam

If the printer runs out of paper or ribbon, you may be able to load a new box of paper or mount a new ribbon without losing or destroying part of the printed output. If so, you will be able to start printing again without using the PROP command. The phantom continues printing where it stopped.

If you encounter a paper jam or other situation where output is lost, use the following procedure:

1. Take the printer off line and attend to the problem, that is, load a new box of paper on the printer or remove a torn page and reset top of form. Prepare the printer to start printing.
2. At this point you can take one of three possible actions
   - Abort printing the current file, but leave the request in the print queue, by issuing the command PROP environment -ABORT. Take this course if there are other requests in the queue that need to be dealt with more urgently than the file that was printing when the problem arose.
   - Reprint the current file from the start, by issuing the command PROP environment -RESTART. Use this option if little of the file has printed.
   - Reprint the current file from a suitable point earlier in the document, by issuing the command PROP environment -BACK n, where *n* is the number of pages that you want to back up. You may need to use PROP -STATUS to find out how many pages had been printed before the paper jam, and so decide how many pages to back up.

You can use PROP -BACK with PostScript printers, and the command works correctly if the laser printer is emulating a line printer, or is producing small facsimile pages on which the layout emulates line printer output.

The -BACK option does not necessarily have a useful effect if the PostScript procedure does complex text formatting. Also, if you must switch off a laser printer, the PostScript procedures must be reloaded before it can continue. In either case you should use the -RESTART or -ABORT options.

APPENDICES

A

SUMMARY OF SPOOL AND PROP COMMANDS

This appendix is for quick reference to the SPOOL and PROP commands, which it describes in turn. For each command there is a brief description of what you can use it for, a summary of its options, a list that gives a full description of each option, and a concise explanation of how you use the command for specific operations.

Spool Command

You can use the SPOOL command to

- Add a request to the spool queue.
- Modify a print request.
- Cancel a print request.
- View the spool queue.

Adding a Request

The command syntax is as follows:

```
SPOOL pathname [ options ]
```

pathname is the pathname of the file to be printed. If the file is in your current directory you can give the filename only. If the file is in a password-protected directory that has a non-blank password, the complete pathname must be given, enclosed in quotes. The password must follow the directory name, as shown in the following example:

```
SPOOL 'MYDIRECTORY PASSWORD>FILE
```

Summary of Options

The American/English options to the SPOOL command are listed below, with brief explanations, for easy reference. For other languages the option abbreviation will be defined in the localized text database. The underscores in the names are stripped out for the American/English text database.

Option	*Meaning*
-ALIAS name **-ALI**	Replace user name in header.
-AS name	Replace file name in header.
-ATTRIBUTE name **-ATT**	Specify device attributes.
-CANCEL n **-CAN**	Cancel a print request.
-COBOL **-COB**	Cobol format.
-COPIES n **-COP**	No of copies (1 - 99).
-DEFER hh:mm **-DEF**	Defer printing to given time.
-DISK name **-DIS**	Place request in pre-Rev. 21.0 queue on named disk.
-ERROR_BRIEF **-ERRB** **-ERROR_MEDIUM** **-ERRM** **-ERROR_DETAIL** **-ERRD**	Control presentation level of Spooler error messages, default is ERROR_MEDIUM.
-FROM m	Print part of a document. See also -TO -FTN Fortran format.
-FTN	Fortran format.
-HEADER xx **-HDR**	Use string *xx* as the page header.
-HELP **-HEL**	Displays details of command syntax.
-LIST **-LIS**	List the spool queue.

-LNUMBERS **-LNU**	Print with line numbers.
-MODIFY n **-MOD**	Modify an entry in the spool queue.
-NO_COPY **-NOC**	Print from original file location.
-NO_EJECT **-NOE**	Inhibit form feed after final page.
-NO_FORMAT **-NOF**	Inhibit all format actions.
-NO_HEADER **-NOH**	Print without header page.
-NOP	Inhibit overprinting.
-NO_SWO **-NOS**	Used via -MODIFY option to cancel the -SPOOL_WHILE_OPEN option.
-NOTIFY **-NFY**	Notify user on completion.
-NO_XLATE **-NOX**	Used via -MODIFY option to remove the -XLATE option.
-NPH	No page header format.
-ON node	Place request in queue on named node.
-OPEN **-OPE**	Open a file unit rather than copying file.
-PLOT n **-PLO**	Plot file.
-PROC name **-PRO**	PostScript procedure name.
-RUSH **-RUS**	Treat as priority request.
-SET_FONT fontname	Selects the typeface font to be used by a printer.
-SET_LANDSCAPE	Text is printed across the longest width of a given size of paper.
-SET_PAPER_BIN n	Selects the paper bin n to be used by a MANUAL printer, or manually feeds the paper.

-SET_PORTRAIT	Text is printed across the shortest width of a given size of paper.
-SFI	Suppress file information in the header.
-SPOOL_WHILE_OPEN **-SWO**	Allows a file to be printed while it is open for writing to disk.
-TO n	Print part of a document. See also, -FROM.
-TRUNCATE **-TRU**	Truncate long lines.
-TUNIT n **-TUN**	Unit number for -OPEN. Default is 2.
-XLATE mapname **-XLA**	Allows selection of alternative mapping to the printer default.

Detailed Descriptions of Options

This section gives full details of how to use each option when you use SPOOL to make a print request.

Option	*Description*
-ALIAS name **-ALI**	The header page at the start of a spooled file -ALI includes a large character representation of the name of the user who submitted the request. The -ALIAS option allows you to replace the name on the header with an alternative string such as the name of another user. *name* is the name that you want to appear in the header. If *name* contains a space you must enclose the whole name in single quotes. Using the -ALIAS option does not prevent your user-ID from being used for other purposes, such as in the accounting data. -ALIAS cannot be used with a request submitted to a pre-Rev. 21.0 queue. Note that the user-ID is not changed and still appears correctly on the header page.
-AS name	The header page includes a large character representation of the file name of the file being printed. This option allows the text of the line to be changed. *name* is the document title that you want to appear. If *name* contains a space, you must enclose it in single quotes.

-ATTRIBUTE att_name [att_name]
-ATT

Specifies device attributes for requests in a Rev. 21.0 queue. You can give one or more attribute names. Alternatively the option itself, -ATTRIBUTE, can be repeated as many times as necessary. The attribute names are checked against a set of valid attribute names supplied by the administrator. You can use -ATT as an abbreviation for -ATTRIBUTE.

In a pre-Rev. 21.0 spool queue, the -ATTRIBUTE option specifies the destination name of the printer. If you give -ATTRIBUTE with no qualifying argument, the despooler looks for an L.DFLT file in the SPOOLQ directory on the specified partition and, if it exists, uses the default destination name held in it.

-CANCEL n
-CAN

Cancels a print request. For details see the section Canceling a Print Request, later in this appendix.

-COBOL
-COB

Specifies that printing should commence in Cobol format.

-COPIES n
-COP

Specifies the number of copies of the file to print. The default is a single copy. When more than one copy is printed, only one header is produced, before the first copy. *n* is the number of copies. The maximum is 99.

-DEFER hh:mm
-DEF

Specifies the earliest time at which a request is to be printed. You may omit the colon between the hours and minutes. It is normally treated as a 24-hour clock time but you can use the suffixes AM and PM to give a 12-hour format. If you leave out the time altogether (that is, just give -DEFER), midnight is assumed. In a network that crosses time zones, the time on the machine holding the queue is used. Formats matching the DATE FULL, UFULL and VFULL are also supported. Strings holding spaces must be quoted.

-DISK ident
-DIS

Specifies the disk partition of a queue in pre-Rev. 21.0 format, to which the request is to be added. To allow existing programs to run unchanged on Rev. 21.0, the SPOOL command ignores the -DISK option if it refers to a disk on the local system. Furthermore, if an attempt to add a request to a remote spool queue fails, and the remote system appears to be running Rev. 21.0, the SPOOL command changes the -DISK option into an appropriate -ON option and tries again. *ident* is a disk partition (MFD) name.

-ERROR_BRIEF
-ERRB
-ERROR_MEDIUM
-ERRM
-ERROR_DETAIL
-ERRD

Control presentation level of Spooler error messages, default is ERROR_MEDIUM.

-FROM m -TO n
-FRO

Enables you to spool part of a document without having to remove unwanted leading and/or trailing pages.

When in normal paginate format this option prints physical page *m* through *n* inclusive, and prints logical pages when in Fortran or Cobol format.

Either -FROM or -TO may be omitted. The default values are from page one to infinity. The -FROM value must not be greater than the -TO value.

Both the -FROM and -TO options work with multiple copies of files using the -COPIES n option.

Note

Embedded specific escape sequences, from pages prior to *m*, may not take effect as the printer will not see them.

-FTN

Specifies that printing should commence in Fortran format.

-HEADER text
-HDR

Replaces the default header. The default is to use the first line of the file. The -HEADER option allows you to replace the default with a header of your own choosing.

text is the header that you want to appear. Any format commands embedded in the file that change the page header still function. If *text* contains spaces, you must enclose the entire text in single quotes. You can give the keyword FILE in place of text, to use the name of the file as the page header.

-HELP
-HLP

Displays help with the command syntax.

-LIST
-LIS

Lists one or more spool queues. For details see the later section, Viewing the Spool Queue.

-LNUMBERS
-LNU

This option prints the line numbers of the lines in the data file. Line numbers precede each line of data in the source file. A data file line that takes up more than one line in the printed output only has a number on the first printed line. Data file lines that overprint the previous line are not numbered.

-MODIFY
-MOD

Changes the options of a print request. For details see the section Modifying a Print Request, later in this appendix.

-NO_COPY
-NOC

The file to be printed is not copied to the SPOOL_DATA* directory. Instead the file is printed from its original file system location. Any changes made to the file between issuing the SPOOL command and printing appear in the printed output. The file must reside on the same node as the spool queue to which the request is added. The despooler phantom that processes the request must be able to attach to the directory that contains the file, and must have read access to the file. You cannot use the -NO_COPY option with requests submitted to pre-Rev. 21.0 queues, nor with password protected directories.

-NO_EJECT
-NOE

This option prevents a form feed after the final page of a file is printed. It is advisable to use -NO_EJECT without headers, and in NO_FORMAT mode.

-NO FORMAT
-NOF

Prints the file in no-format mode. This option is used when printing a PostScript file on a PostScript printer. It cannot be used with the -PLOT option.

-NO_HEADER
-NOH

Prints the file without header pages. Note that the -NO_HEADER option is not the converse of -HEADER.

-NOP

Inhibits overprinting, whether required by the FORTRAN or COBOL format + control character, or by trailing carriage return characters. The Prime convention is to terminate lines with a line feed character only. The -NOP option is useful with output that uses a carriage return and a line feed as the line terminator.

-NO_SWO
-NOS

This option, used through the -MODIFY option, cancels the -SPOOL_WHILE_OPEN process. The despooler treats the file as if it is to be printed with the -NO_COPY option, and the special end of file (EOF) handling is dropped.

-NOTIFY
-NFY

When the file is printed a message is sent to the user who made the request. This option cannot be used with the -ON option. If this is attempted an error message will be displayed.

-NO_XLATE
-NOX

Used via -MODIFY option to remove the -XLATE (alternative mapping) option.

-NPH

No page header and no page numbers are printed. Note this is not the same as -HEADER, or -NOH.

-ON node

Allows you to add a request to a Rev. 21.0 or higher, spool queue on a specific node of a network. *node* is the node name.

-OPEN
-OPE

The -OPEN option opens a data file in the SPOOL_DATA* directory. Data is generated by a program and directed to the data file. Unless otherwise specified (see -TUNIT) the file is opened on unit 2.

-PLOT n
-PLO

Specifies the plot raster size for a plotter output file. This option does not send a file to a plotter. To select a plotter use -ATTRIBUTE with the correct attributes for your system. The print mode will be PLOT format though. This option cannot be used with the -NO_format option. *n* is the raster size in words per scan. Default is 128.

-PROC name
-PRO

Specifies the PostScript procedure to use to print a document on a PostScript laser printer. The procedure must be one that was loaded when the printer initialized. If the procedure is no longer loaded, which could happen if the power supply to the printer had been interrupted, then it is reloaded.

-RUSH
-RUS

Gives priority to a request, within its queue. That is, marks a request as having priority over non-rush requests in the same spool queue. The request is selected for printing without reference to size and defer time restrictions. You must be a privileged user to use -RUSH.

-SET_FONT fontname

Selects the font typeface to be used by a printer. *fontname* is a character string that has a maximum length of 32 characters.

-SET_LANDSCAPE

Selects landscape paper orientation. For a given size of paper a printer is printing landscape if the text is printed across the longest width.

-SET_PAPER_BIN $\left\{\begin{array}{l} \text{n} \\ \text{MANUAL} \end{array}\right\}$

Selects the source of the paper to be used by the printer. Some laser printers have more than one paper bin, and *n* represents the range of paper bins 1-9. MANUAL specifies that paper is fed manually.

-SET_PORTRAIT

Selects portrait paper orientation. For a given size of paper a printer is printing in portrait if the text is printed across the shortest width. This document is printed in portrait orientation.

-SFI

Suppress File Information. The file pathname, and the date and time the file was modified, are not printed in the header or trailer pages. Useful where the information is considered to be confidential.

-SPOOL_WHILE_OPEN
-SWO

This option allows a file to be printed while it is open for writing to disk. The file is not copied into SPOOL_DATA* when the -SWO option is used. Without -SWO selected the despooler terminates when it reaches the end of file (EOF). With the -SWO option, the despooler waits for a defined time, and checks a number of times before it terminates: the Wait Time, and Times to Check parameters are defined in the printer environment definition file. Note that -SWO does not guarantee immediate printing, but only starts printing when a printer with the correct attributes is available.

-TRUNCATE -TRU		Truncates lines that are longer than the width of the maximum printer line, as defined in the environment file. If this option is not given, long lines are wrapped round two or more lines so that no text is lost. Where this occurs in a file that uses Fortran or Cobol format modes, a request to overprint a line that has been split is ignored. Instead the overprint data is printed as a separate line.
-TRUNIT n -TUN		Specifies the file unit associated with the -OPEN option. *n* is the file unit number. If you do not use -TUNIT, the default file unit number is 2.
-XLATE -XLA	[mapping]	This option allows you to select an alternative mapping to the printer's default. *mapping* is a 1-32 character string you use to select the mapping. The first character is alphabetic, and the remainder, alphanumeric, including . $ or _.

Modifying a Print Request

If you are a privileged user you can modify any request in the queue, otherwise you can modify only your own requests. The command to modify a request in the spool queue is as follows:

```
SPOOL -MODIFY request [ options ]
```

where *request* is the request number.

You can find the request number of an entry in the spool queue by viewing the queue with SPOOL -LIST, as described later in this appendix.

All the options available when you add a request to the spool queue can be used except -NO_COPY, -OPEN, and -TUNIT. Generally speaking, modifying a request adds an option. If the new option conflicts with a previous option, the previous option is removed. For example, specifying -FTN would cancel an existing -NO_FORMAT in the original request.

If you change the attributes of a request with the -ATT option, you must specify all the attributes that the changed request is to have.

To cancel a -DEFER or a -RUSH option, use -NO_DEFER and -NO_RUSH respectively.

Canceling a Print Request

To cancel a request you must know its request number. You can find the request number of an entry in the spool queue by viewing the queue with SPOOL -LIST, as described later in this appendix. If you are a privileged user you can cancel requests belonging to other users by request number, otherwise you can cancel only your own requests.

The command to cancel print requests is as follows:

```
SPOOL -CANCEL { request } [ -ON node   ] [ options ]
              { -ALL    } [ -DISK name ]
```

request is the request number. If you give the keyword -ALL in place of a request number, all your requests are canceled. If you are a privileged user, you cannot use -ALL to cancel other users' print requests. To cancel another user's request you must give the request number.

node is a node name.

name is a pre-Rev.21.0 disk name.

Viewing the Spool Queue

Privileged users see the entire queue on Rev.21.0 nodes and later other users can only see their own requests. This restriction can be relaxed by the administrator as described in Chapter 2, Setting Up The Spooler Subsystem.

The command to view the spool queue is as follows:

```
SPOOL -LIST [ request ] [ options ]
```

Summary of Options

The available options are listed, with brief explanations, in the following table.

Option	*Meaning*
-ALL	List all queues as defined by the system administrator in the file SPOOL* > QUEUES.
-ATT name ...	Report only entries with given attributes. name is an attribute name.
-BRIEF -BR	Produces a short report. This is the default option and is the same as SPOOL -LIST
-DETAIL -DET	Produces a more detailed report.
-FULL	Gives the same report as -DETAIL, plus, for each print request, the name of the partition that holds the copy of the file to be printed.
-USER name	Reports only requests for user-ID name
-ON node	Report on queue on the named node. *node* is the node name.

-DISK name — Report the pre-Rev.21.0 queue on the named partition. *name* is a disk partition (MFD) name.

-NO_WAIT
-NW — Suppress the `--More--` prompt if the display takes up more than one screen.

Four levels of report are available. The -DETAIL option provides an intermediate level of information. The report produced by the SPOOL -LIST -DETAIL command is of the following format:

```
System SYSA
Request   Time   User                    Copies  Size      State
--------  -----  ----------------------  ------  -----     -----
7         10:06  SMITH                   3       99        (66% Printed)
File name   <DISKA>DOCS>REPORT.RUNO
Attributes  DOC, SYSA
Options     -XLATE FILENAME
Despooler   DOC on node: SYSA

8         10:07  SMITH                   1       118
File name   <DISKA>PLOTTER>DIA_173
Attributes  PLOTTER
Options     -NO_COPY -PLOT 224

9         10:09  SMITH                   1       55        (Printing)
File name   <DISKA>ACCTS>REPORT.5
Attributes  DOC
Options     -SPOOL_WHILE_OPEN -NO_COPY -NOTIFY -XLATE
Despooler   DOC on node: SYSC
```

For each request, the report always shows details of the submitting user, time of submission, number of copies, total size and file pathname. Even if the -SFI option is used in a print request the file pathname appears in the display produced by SPOOL -LIST -DETAIL.

The attributes line shows any device attributes in the request, both those explicitly given by the user and those supplied as defaults. The line is only present if the request includes at least one attribute.

The options line shows the options used in the spool command. It is only present if at least one option is to be reported.

The defer time line shows the earliest time at which the file will be printed.

The Printing tag appears against any request currently being printed. The % Printed tag is calculated in records, not pages.

If the report includes requests from an earlier date, the date is included before the first reported request and at each point where it changes.

The -FULL option adds extra information probably only of use to administrators, and then only infrequently. The additional data is a single line that identifies which partition holds the SPOOL_DATA* directory containing the copy of the data file.

The -BRIEF option produces a short report with a single line per request, and is the default. This report gives a general picture of the queue without all the detail. The same queue as in the above example would appear as follows:

```
System MFH
Request Time  User      File              No Size  State
------- ----  --------- ----------------- -- ----- -----
7       1008  SMITH     REPORT.RUNO       3  99    Print
8       1007  SMITH     DIA_173           1  118
9       1009  SMITH     REPORT.5          1  55    Print
```

Configurable SPOOL -LIST [-BRIEF] displays are explained in Chapter 2, Viewing the Spool Queue.

Prop Command

The name PROP is an acronym for PRinter OPeration. The PROP command allows you to control despooler phantoms, to view the status of despooler phantoms that have been started, and to check the correctness of new environment files. For details of environment files see Chapter 3, Defining Printer Environments.

The syntax of PROP command is as follows:

```
PROP   { environment option }
       { -STATUS [-ALL]     }
```

environment is an environment name.

All users can give the PROP -STATUS command and the PROP env -DISPLAY command. However you must be a privileged user to use all other options of the PROP command.

Summary of Options

The available options are listed below with brief explanations, for easy reference.

Option	*Meaning*
-ABORT **-AB**	Postpone current print file.
-BACK n	Back-up *n* pages.
-COLDSTART **-COLD**	Initialize SPOOLER environment.
-CONTINUE **-CONT**	Resume printing after a PROP -HANG, or a PROP -LINEUP.

-DISPLAY **-DISP** **-DP**	Show status and environment file of named environment.
-DROP	Abandon current print file.
-ENV_LANGUAGE lang-name **-ELANG**	Defines the environment file language.
-HANG	Suspend a despooler phantom.
-HELP **-H**	Displays details of command syntax.
-LINEUP nn **-LU**	Paper alignment check (*nn* lines).
-NO_WAIT **-NW**	Suppress --More-- prompt.
-RELEASE **-REL**	Releases a suspended printing job.
-RESET [new_env]	Refresh a despooler environment.
-RESTART **-RES** **-RST**	Restart current print file.
-START	Start up a despooler phantom.
-STATUS **-STAT**	Display status of environments.
-STOP	Stop a despooler phantom.
-SUSPEND **-SUS**	Suspend a currently printing job.
-VERIFY **-VFY**	Verify an environment file.

Detailed Descriptions of Options

This section gives details of how to use each option of the PROP command.

Option	*Description*
-ABORT **-AB**	Aborts printing the current job without removing the request from the spool queue.

-BACK n

Backs up a given number of pages and reprints the data. Use this option after a paper wreck or other problem with the printer. *n* is the number of pages, and has a default of 1.

-COLDSTART
-COLD

Used during machine cold start to initialize the data structures used by the Spooler subsystem, and to reset the queue control information. (See Chapter 2 for cold-start procedures.)

-CONTINUE
-CONT

Restarts printing after using -HANG or -LINEUP.

-DISPLAY
-DISP
-DP

Displays status of named environment and lists the environment file.

-DROP

Aborts the currently printing job and deletes the request from the spool queue.

-ENV_LANGUAGE lang-name
-ELANG

The Standard Internationalization Tool (SIT) supported language of the environment file, only useful on multi-lingual sites.

-HANG { -NOW | -IDLE | -FINISH | -FIN }

This option makes the despooler phantom pause, to allow the operator to attend to the printer, for example to change the ribbon. Use the -CONTINUE option to restart. The -NOW, -IDLE, and -FINISH options determine when the action is to take place. Use -NOW to suspend the despooler immediately, even if a file is currently being printed. -FINISH suspends the despooler when it has printed the file. -IDLE suspends the despooler when it becomes idle.

-HELP
-H

Displays details of command syntax.

-LINEUP nn
-LU

The -LINEUP option backs up the despooler to the start of the current page, prints *nn* lines of the document, then pauses and displays the message 'paper alignment check'. Printing is restarted by a further PROP command with either the -CONTINUE option (which does not reprint the first part of the data) or the -LINEUP option to repeat the line-up sequence.

-NO_WAIT
-NW

Suppress the `--More--` prompt if the display takes up more than one screen.

-RELEASE { -FINISH | -FIN | -IDLE }
-REL

This option releases a previously suspended printing job, and restarts it from the top of the page it stopped at. The -FINISH and -IDLE options determine when the action is to take place. -FINISH releases the despooler when it has printed the current file. -IDLE releases the despooler when it becomes idle.

-RESET [newenv] $\left\{\begin{array}{l}\textbf{-NOW}\\ \textbf{-IDLE}\\ \textbf{-FINISH}\end{array}\right\}$

Stops and restarts a despooler phantom. Useful if the environment file has been altered. The optional new environment name, *newenv*, allows you, with a single command, to replace one despooler environment with another. The -NOW, -FINISH and -IDLE options determine when the action is to take place as described under -HANG.

-RESTART
-RES
-RST

Restarts the current print job.

-START

Starts the named environment, provided that you have access to the environment file.

-STATUS [-ALL] [-NO_WAIT]
-STAT

Displays a one-line status message for each active environment. If the despooler phantom is printing a request the progress of the job is shown. If you use -ALL, the status is displayed for all environments for which environment files exist. The -NO_WAIT option inhibits the normal pagination of the display.

-STOP $\left\{\begin{array}{l}\textbf{-NOW}\\ \textbf{-IDLE}\\ \textbf{-FINISH}\end{array}\right\}$

Shuts down the named environment. The further options -NOW, -FINISH,and -IDLE specify when the action is to take place. Use -NOW for immediate shut down, even if a file is currently being printed. -FINISH shuts down the despooler when it has printed the file. -IDLE shuts down the despooler when it becomes idle.

-SUSPEND
-SUS

Suspends the current printing job.

-VERIFY

Checks the syntax of an environment file and notifies user of any errors.

B

HOW TO WRITE A PRINT HANDLER

Print Handlers

A print handler is a piece of software used by a despooler phantom to perform standard output operations and other related device handling for a particular type of output device. It is not user visible.

Advantages of Print Handlers

The print handler is entirely separate from the despooler. Separated in this way, all output devices appear the same from the viewpoint of the despooler phantom, and it can use similar output control commands for all the print handlers. Consequently, you can support any new output device by writing a suitable print handler. There is no need to make any change to the despooler code.

Print Handlers Provided by Prime

Print handlers are provided for the following devices:

DATAPRODUCTS
DIABLO
DPTX
NEC
QUME
QUME_OAS
POSTSCRIPT
PRINTRONIX
TTY
QMS
NEC_OAS

The code for the print handlers is located in the file SPOOL*>DESPOOL_LIBRARY.RUN, which is a library class EPF.

Specifying Which Print Handler to Use

To specify the print handler for a despooler, use the DEVICE command in the environment file, as explained in Chapter 3, Defining Printer Environments.

How To Write A Print Handler

To create a new print handler you need technical information about the relevant device. You also need to know

- An efficient way of organizing the work
- How the despooler software is organized
- What functions the print handler must provide
- What low level drivers are available to you
- Details of the interfaces with DESPOOL.RUN and useful subroutines
- How to build a print handler
- Details of the data structures that you must use

Each of these is described in a separate section below.

Organizing the Work

The despooler and the print handler software supplied by Prime are coded in PL1, therefore it is recommended that you also work in this language. In addition to the standard insert files in SYSCOM, PL1 insert files are provided in the SPOOL*>SOURCE directory. Some support is given for FTN so this would be a sensible second choice. You can use other languages, but this will entail more work.

When you have coded and compiled a new print handler you should then test it. To test a new print handler, create a separate EPF library, and modify the search rules in SPOOL*>ENTRY$.SR so that a despooler phantom can find it. When the print handler is fully tested you should add it to the despooler search rules.

See the *Programmer's Guide to BIND and EPFs*, to find out how to create EPFs with the BIND command.

The source code should follow the pattern below.

Procedure label and declaration
Standard insert files from SPOOL*>SOURCE
Standard insert files from SYSCOM
Required data declarations
Replacement strings
Further data declarations
Body of program

You should construct the body of the program as a select clause, that deals with each possible print handler function in turn.

See the section entitled Example of a Print Handler for an example that you can use as a model when you construct your own print handlers.

How the Despooler Software is Organized

The structure of the despooler software is summarized in Figure B-1. SPOOL*>DESPOOL.RUN is a program class EPF. It is invoked for each despooler phantom that is started. The PRINT_FILE routine within DESPOOL.RUN issues print directives to the appropriate print handler in SPOOL*>DESPOOL_LIBRARY.RUN.

The print handler in turn makes use of low-level device drivers that are part of DESPOOL.RUN. These low-level device drivers provide a common interface to the PRIMOS device/line driver subroutines.

Figure B-1 also shows SP$FLP, which is the entry point for a routine that print handlers use to convert ECS data to ASCII-8

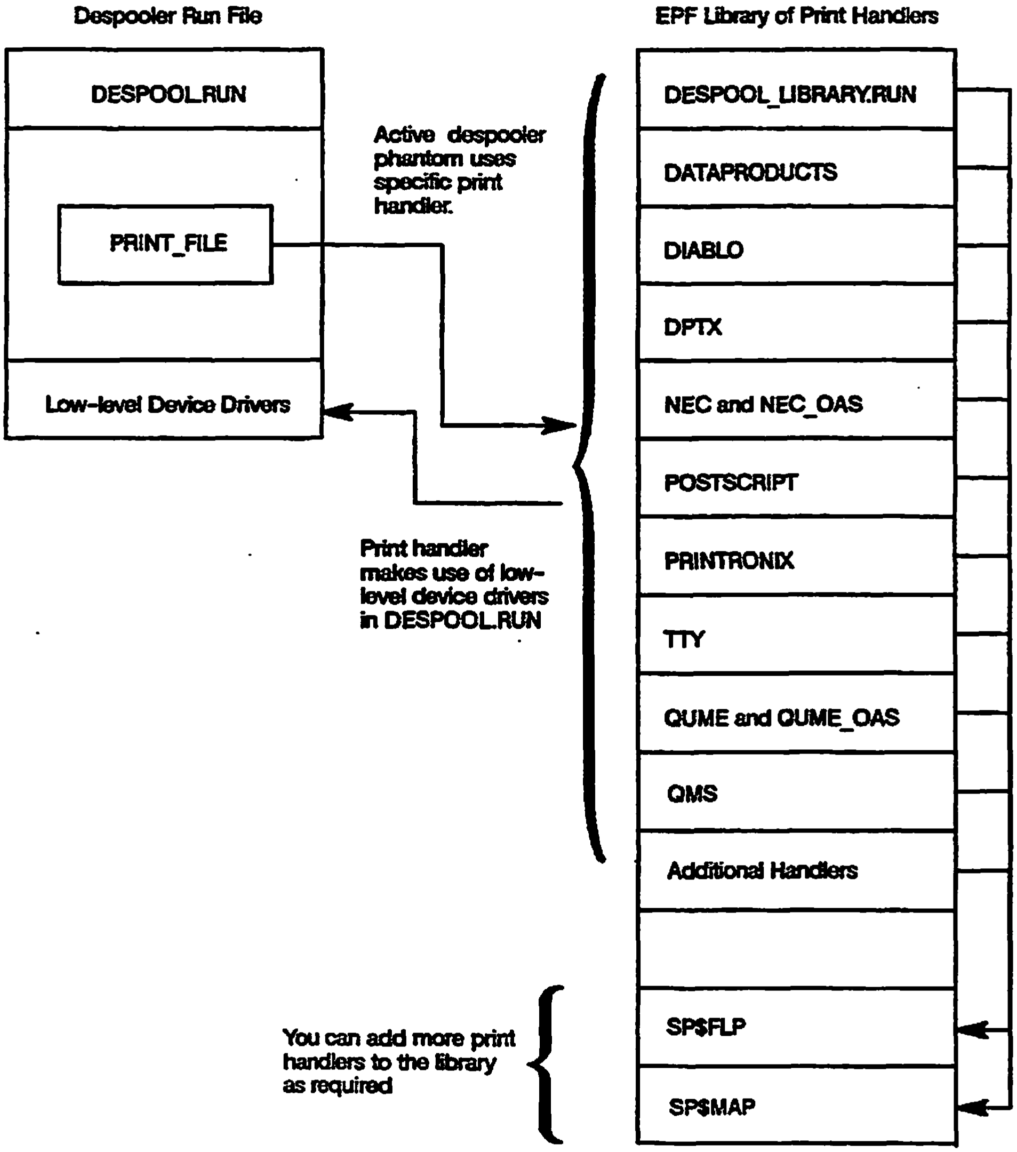
Despooler Run File
DESPOOL.RUN
PRINT_FILE
Low-level Device Drivers
Active despooler phantom uses specific print handler.
Print handler makes use of low-level device drivers in DESPOOL.RUN
You can add more print handlers to the library as required
EPF Library of Print Handlers
DESPOOL_LIBRARY.RUN
DATAPRODUCTS
DIABLO
DPTX
NEC and NEC_OAS
POSTSCRIPT
PRINTRONIX
TTY
QUME and QUME_OAS
QMS
Additional Handlers
SP$FLP
SP$MAP

FIGURE B-1. Structure of the Software

Functions Provided by a Print Handler

Each print handler must provide code for the functions listed below. The despooler phantom expects all the functions to be present. You must therefore provide an entry for each function, even if the code only performs an immediate return to the calling routine.

The key names are the names that the despooler phantom uses when it requests an action from a print handler.

Key — *Function*

S$BEEP_ — Sends code or escape sequence to output device to generate an audible tone.

S$BIN_ — Tells the device handler to change to the paper bin that is passed as the argument.

S$END_DOC_ — Processes the end of the user document. For example, reverts to standard font to print trailing pages.

S$END_JOB_ — Performs end of job processing. For example, resets output device characteristics.

S$FONT_ — Tells the device handler to change to the font that is passed as the argument.

**S$INIT_** — Initializes the output device. Usually this means setting characteristics of the output device and the line to which it is connected. The characteristics initialized here are assumed to last until the despooler phantom that controls the device is logged out. Some examples of tasks that may need to be performed by S$INIT are: assign a line to the output device, download a special file, and clear font memory. The particular tasks, and precise details of the initialization, depend on the device and the way that your system is organized.

S$INPUT_ — Reads input from a device and returns the input string to the calling program.

S$LARGE_PRINT_ — Formats and outputs a message in large characters.

**S$MAKE_READY_** — Brings the printer to a state of readiness to print, that is, to a state in which the S$START_JOB_ command works.

S$PAGE_ — Sends the appropriate code or escape sequence to the device to make it form feed.

S$PORT_ — Tells the device handler to change to either portrait or landscape.

S$PRINT_ Accepts a line of print passed in a buffer, translates it to the correct characters for output, passes the output string to the correct line driver, and updates the line count in the PAGCOM data structure.

S$START_COPY Sent at the start of each copy of a document. For a single copy it is sent once, for multiple (*n*) copy documents it is sent once per copy (*n* times).

S$START_DOC_ Initializes the start of a user document; for example, selects the correct font, if different from the header page.

S$START_JOB_ Initializes each job, for example, checks that procedures are loaded in a PostScript laser printer.

**S$START_PRINT_** Sends a code or escape sequence to make the device continue operation. This key is used after S$STOP_PRINT_ has been called.

S$STATUS_ Reads status from the device, and returns it to calling program.

S$STOP_PRINT_ Sends a code or escape sequence to stop the output device or make it pause.

**S$UNREADY_** Returns the device to the state before S$MAKE_READY was executed.

S$VALIDATE_ Not implemented. You must provide a dummy entry and an immediate return for this function.

In addition to providing the above functions the print handler must convert the input character set to the printer character set. See SP$FLP and SP$MAP subroutines that are described in the later section of this chapter, Details of Interfaces.

Sequence of Key Names to a Print Handler

A print handler expects to receive the key names in the following sequence:

```
S$INIT_
when a despooler is started
  S$MAKE_READY_
  when work is available
    S$START_JOB_
    print header pages
    S$START_DOC_
      S$START_COPY_
      print all copies, s$start_copy_ per copy
    S$END_DOC_
    print trailer pages
    S$END_JOB_
S$UNREADY_
when no more work available
```

Low-Level Device Drivers

Print handlers should invoke low-level device/line drivers through the common interface provided by DESPOOL.RUN, irrespective of the type of line or device that is used. They should not contain device specific code. The functions provided by DESPOOL.RUN deal with connecting to and disconnecting from the device, and handle all communication with the device. DESPOOL.RUN provides the following low-level device/line drivers:

- Asynchronous Lines
- Parallel Lines
- Public Data Networks
- Terminal Lines
- Disk Files
- IBM devices connected by Distributed Processing Terminal Executive (DPTX)
- Printers connected to a TCP/IP network

Further information is given in the following section, Details of Interfaces.

Details of Interfaces

This section describes the following interfaces:

- From DESPOOL.RUN to the print handler
- From a print handler to low-level device/line drivers in DESPOOL.RUN
- From a print handler to the character conversion subroutines

Calling Interface from DESPOOL.RUN to the Print Handler

The call (in PL1) is as follows:

```
call print_handler (key, buffer, qcom, envcom, pagcom, code);
```

where *print_handler* is the name of the print handler, for example QUME or DATAPRODUCTS, and the parameters are as shown below:

Item	*Type*	*Description*
key	fixed bin(15)	Name of print handler function to invoke
buffer	char(1024) var	Text buffer
qcom	1 (structure)	See the later section, Data Structures
envcom	1 (structure)	See the later section, Data Structures
pagcom	1 (structure)	See the later section, Data Structures

code	fixed bin(15)	Returned error code

Calling Interface from Print Handler to Low-level Device/Line Drivers

The call (in PL1) is as follows:

```
call addr(envcom) -> envcom_based.sp$dtr (key, bp, nc, instr, code);
```

Item	*Type*	*Description*
key	fixed bin(15)	Name of device driver function to invoke
bp	ptr	Pointer to user's data buffer
nc	fixed bin(15)	Number of characters to input or output
instr	fixed bin(15)	Parallel instruction word
code	fixed bin(15)	Returned error code

Key	*Function*
S$ASGN	Assign device, and initialize device driver.
S$UNAS	Unassign device, and undo initialization of device driver.
S$MRDY	Bring device to ready to print state.
S$URDY	Return device to condition before S$MRDY was executed.
S$STRT	Start transmission of file to be printed.
S$ENDT	End transmission of file to be printed.
S$GETB	Read buffer from device.
S$PUTB	Write buffer to device.
S$LSTA	Test status of line/physical connection.
S$BSTA	Test status of device buffer.
S$CLIB	Clear input buffer.
S$CLOB	Clear output buffer.

Calling Interface from Print Handler to Character Conversion Routines

Before each call to a low level device driver you need to ensure that the output is consistent with the character set of the printer. There are two routines provided to achieve this. You can call either SP$FLP or SP$MAP.

SP$FLP converts ECS data to ASCII. This ensures that ASCII is consistently sent from your print handler to the output device. The subroutine can be called from either PL1 or FTN.

In PL1 you should declare the entry point as follows:

```
dcl sp$flp entry (char(*) var) options (shortcall);
```

The PL1 call is as follows:

```
call sp$flp (ECS_data)
```

In FTN you must declare the entry point in a SHORTCALL statement, as follows:

```
SHORTCALL SP$FLP
```

You must then use the compiler option -SPECIAL_OPTION (-SPO), otherwise the SHORTCALL statement is not recognized.

The parameter that is passed to SP$FLP is of type INTEGER*2, and must be declared as such. The calling sequence is therefore as follows:

```
INTEGER*2 P8DATA(1025)
     .         .
     .         .
     .         .
CALL SP$FLP (P8DATA)
```

SP$MAP is a general purpose routine capable of converting characters from one character set to another.

In PL1 you should declare the entry point as follows:

```
dcl sp$map entry (ptr, char(1024) var, bin, char(1024) var, bin, bin);
```

The PL1 call is as follows:

```
call sp$map(envcom.map_table_ptr, source_string, current_position, target_string,
            current_status, status);
```

Call — *Function*

envcom.map_table_ptr — Is the pointer to the map as provided in the environment data area.

source_string — Is the varying string to be converted.

current_position Is the point (within source_string) at which to start, or continue, the conversion.

target_string Is the converted string.

current_status Is either zero or the last value of status if the source string requires more than one call to complete the conversion.

status Indicates the result of the conversion. If 0 then the conversion is complete. If greater than zero then an unrecoverable error occurred during the conversion, the other returned variables should be ignored. If less than zero, then the source string could not be completely converted in this call.

It may be necessary to call SP$MAP more than once for a particular line of output. When this happens status is returned as a negative value. After you output the portion of the converted line, returned in target_string, you should call SP$MAP again. When you call again, the current_position should be the value returned by the last call to SP$MAP, and current_status should be value of status from the last call.

If status is less than -999, that is -1000, then the return buffer, target_string, is not large enough. The handler should output the target_string that has been returned and then hold the print position at the end of it. Thus the next piece is printed at the end of the first. If status is in the range -1 through -999, then the line contains characters that must be built up by overprinting. The returned target_string should be output and the print position returned to the beginning of the current line.

Auxiliary Configuration Parameters

You can use the AUXILIARY command in the printer environment files to pass environment parameters to your self-written print handlers and routines. The environment parameters are read from the environment file, and are inserted into an auxiliary structure in external common memory, from where they are accessed by the self-written user routines. Note that the auxiliary structure is not used by the standard despooler, but exists solely for the benefit of routines written by the user. The content and use of the environment parameters are the users' responsibility.

For further details of the AUXILIARY environment command, see Chapter 3, Defining Printer Environments.

How to Build a Print Handler

The following example illustrates the procedure to build your print handler.

```
OK, BIND
: LIBMODE -PROGRAM
: LOAD Routine
: LIBRARY
: DYNT SP$FLP
: DYNT -ALL
BIND COMPLETE
: FILE
OK,
```

Data Structures

Three data structures used by DESPOOL.RUN are accessible to the print handler. Each structure provides information needed to perform some of the print functions correctly. You must keep some of the data items up to date, but you are free to alter others if necessary.

QCOM: This data structure holds information about the entry in the spool queue. The declaration is as follows:

```
dcl 1 qcom_based based,
  2 flags,
    3 printing_from_pre_rev_21_queue bit(1),
    3 rfu bit(15),
  2 qno bin(31),                                /* Queue entry number */
  2 pp_no bin,                                  /* Poll point number of node or disk */
  2 queue_name char(32) var,                    /* Node or disk name */
  2 qinf,                                       /* Information from the queue entry */
    3 user char(32),                            /* Primos user name */
    3 node char(32),                            /* Node of origin */
    3 file char(180),                           /* File name */
    3 header char(32),                          /* File name for header */
    3 alias char(32),                           /* User name for header */
    3 copies bin,                               /* No of copies */
    3 rqtime char(15),                          /* Request time YYMMDDHHMMSSsss */
    3 dtm bin(31),                              /* Date/time modified */
    3 flags1,
      4 fortran_format bit(1),                  /* Fortran format control */
      4 page_header_supplied bit(1),
      4 numbers bit(1),                         /* Generate line numbers */
      4 no_header bit(1),                       /* Suppress header page */
      4 no_final_eject bit(1),                  /* No page throw at end */
      4 no_format bit(1),                       /* No format control */
      4 plot_file bit(1),                       /* Plot file */
      4 open_option_used bit(1),                /* Set if -OPEN was used */
      4 no_header_format bit(1),                /* -NOPAGEHEADER */
      4 suppress_file_info bit(1),              /* -SFI */
      4 rfu_2 bit(2),
      4 cobol_format bit(1),                    /* Cobol format control */
      4 no_overprint bit(1),
      4 notify_user bit(1),
      4 no_copy bit(1),
    3 flags2,
      4 from_page bit(1),                       /* -FROM option used */
      4 to_page bit(1),                         /* -TO option used */
      4 swo bit(1),                             /* -SWO option used */
      4 rfu_1 bit(3),
      4 truncate bit(1),                        /* Truncate long lines (-TRUNC) */
      4 non_default_xlate bit(1),               /* Use a non default map */
      4 portrait_flag bit(1),                   /* -SET_PORTRAIT */
      4 landscape_flag bit(1),                  /* -SET_LANDSCAPE */
      4 paper_bin_flag bit(1),                  /* -SET_PAPER_BIN */
      4 font_flag bit(1),                        /* -SET_FONT */
      4 rfu_2 bit(4),
    3 ps_proc char(32) var,                      /* PostScript procedure name */
    3 raster_size bin,                           /* Words per raster in plot file */
    3 user_no bin,                               /* User no */
    3 project_id char(32) var,                   /* Project name */
    3 file_size bin,                             /* Size in records * copies */
    3 no_of_attributes bin,                      /* No of attributes following */
    3 attribute(32) char(16),                    /* Device attributes */
    3 page_header char(160) var,                 /* Alternate page_header */
    3 user_address char(32),                     /* From SP$ADDRESS */
    3 from_page_num bin(31),                     /* From page number */
    3 to_page_num bin(31),                       /* To page number */
    3 map_name char(32) var,                     /* Name of map to be used */
    3 words_printed bin(31),                     /* Words printed so far */
    3 file_size_copy bin,                        /* Copy of file_size */
    3 paper_bin bin,                             /* number of bin to use */
    3 font char(32);                             /* name of font to use */
```

PAGCOM: This data structure holds information that the print handler needs to format each page correctly; it also records the current state of the page that is being printed. Your print handler must keep the line counts up-to-date. The declaration is as follows:

```
/* --- Physical definition for format modes; ie the equivalent values that would appear in
       standard print file, after the ^001/^201 control code. */

%Replace no_fmt_ by 0,
         fortran_fmt_ by 1,
         cobol_fmt_ by 2,
         paginate_fmt_ by 3,
         no_header_fmt_ by 4,
         raster_fmt_ by 5,                       /* Aka "binary" mode. */
         vector_fmt_ by 6;                       /* Aka "high-bit-off" mode. */

%Replace max_header_ln_ by 160,
         max_tab_stops_ by 128;

%Replace pagcom_size_ by 235;                    /* Size of PAGCOM in words. */

dcl 1 pagcom_based based,
   2 print_mode bin,                             /* Print format mode. (physical). */
   2 chars_pline bin,                            /* Characters per line. (physical). */
   2 char_count bin,                             /* Character position on current line. */
   2 total_char_count bin(31),                   /* Total number of characters printed. */
   2 lines_ppage bin,                            /* Lines per page. (physical). */
   2 line_count bin,                             /* Current line number on page. */
   2 virt_line_count bin,                        /* Virtual line number on page. */
   2 total_line_count bin(31),                   /* Total number of lines printed. */
   2 page_count bin,                             /* Current page number of this copy. */
   2 virt_page_count bin,                        /* Virtual page number of this copy. */
   2 total_page_count bin(31),                   /* Total number of pages printed. */
   2 header char(max_header_ln_) var,            /* Current header message. */
   2 left_margin bin,                            /* Left margin. (logical). */
   2 right_margin bin,                           /* Right margin. (logical). */
   2 top_margin bin,                             /* Top margin. (logical). */
   2 bottom_margin bin,                          /* Bottom margin. (logical). */
   2 cpi bin,                                    /* Characters Per Inch. */
   2 lpi bin,                                    /* Lines Per Inch. */
   2 tab_interval bin,                           /* Tab Interval. */
   2 num_tab_stops bin,                          /* Number of tab stop positions. */
   2 tab_stops(max_tab_stops_) bin,             /* Top Stop positions. */
   2 pad_char char,                             /* Padding char, space/null/0 etc. */
   2 fmt_flags,                                 /* Extra Format flags. */
     3 append_lf bit(1),                        /* Append LF to print line. */
     3 line_wrap bit(1),                        /* Wrap line after right margin exceeded. */
                                                /* '1'b = wrap, '0'b = truncate. */
     3 pad_char_flag bit(1),                    /* '1'b = pad char specified in option */
                                                /* '0'b = use pad_char in driver */
     3 mbz bit(5),                              /* Reserved for future use */
                                                /* Pre-rev21 sets mbz to zeros */
   2 file_position bin(31),                     /* Saved file position for Backup. */
   2 rev_22_bits,
     3 print_it bit(1),                         /* true if printing this part of file */
     3 to_point_reached bit(1),                 /* true if -TO page number completed */
     3 override_from_to bit(1),                 /* temporarily ignoring -FROM/-TO */
     3 rfu bit(13);                             /* reserved for future use */
```

ENVCOM: This data structure provides the print handler with information about the despooler environment. The declaration is as follows:

```
/* ---- Print handler classes. */

%Replace BUILTIN_ by 1,
         DYNAMIC_ by 2,
         UNKNOWN_ by 3;

/* ******************************************************************************** */

/* ---- Table of builtin handler names. */

%Replace PRINTRONIX_ by 1,
         QUME_ by 2,
         DIABLO_ by 3,
         MAX_TBL_SZ_ by 1;

/* dcl int_hdlr_tbl(MAX_TBL_SZ_) char(32) var static init
                    ('PRINTRONIX', 'QUME', 'DIABLO'); */

dcl int_hdlr_tbl(MAX_TBL_SZ_) char(32) var static init('');

/* ******************************************************************************** */

%Replace MAX_EVFU_ by 143;                      /* Max size of evfu storage. */

%Replace MANUAL_ by 1,                          /* Stationery feeding types. */
         CUTSHEET_ by 2,
         CONTINUOUS_ by 3;

%Replace LOG_OFF_ by 0,                         /* ENVCOM.LOG_TYPE values */
         LOG_ON_ by 1,
         LOG_SAVE_ by 2;

/* ************************************ Envcom ************************************* */

dcl 1 envcom_based based,
    2 handler_name char(32) var,                /* Printer handler name. */
    2 print_handler entry(bin, char(*) var, 1, 1, 1, bin) variable,
                                                /* The printer handler to use. */
```

```
2 big_letters entry(char(*) var, (7) char(*) var, char(*)) variable,
  2 read_evfu entry(1) variable,
  2 page_throw entry() variable,
  2 sp$dtr entry(bin, ptr, bin, bin, bin) variable,
  2 prop_user_no bin,                       /* User no of process that invoked PROP. */
  2 device_type bin,                        /* Low level device driver index. */
  2 comms_ptr ptr,                          /* Address device driver data area. */
  2 gen_flags,                              /* No generic class for these bits. */
    3 evfu_sw bit(1),                       /* Please dll evfu file (below). */
    3 font_sw bit(1),                       /* Ditto for font. */
    3 dict_sw bit(1),                       /* Ditto for dict. */
    3 crlf_sw bit(1),                       /* Auto line feed switch. */
    3 uc_sw bit(1),                         /* Upper case switch. */
    3 ksr_sw bit(1),                        /* Defines printer to be KSR type. */
    3 warning_sw bit(1),                    /* En/Dis-able warning banner. */
    3 evfu_disabled_sw bit(1),              /* Evfu not enabled in hardware. */
    3 rbc_sw bit(1),                        /* Use rolling banner char for
                                               big letters. */

    3 facing_banner_sw bit(1),              /* Select upwards facing banner. */
    3 banner_eject bit(1),                  /* Eject before printing banner */
    3 suppress_file_info bit(1),            /* Suppress file info on banner page */
    3 mbz bit(4),                           /* Must be zero. */
  2 evfu_file char(128) var,                /* EVFU pathname. */
  2 font_file char(128) var,                /* Font pathname. */
  2 dict_file char(128) var                 /* Dictionary pathname (PS printers). */
  2 message_text char(160) var,             /* Msg to display on header banner. */
  2 printer_info(2) char(160) var,          /* Optional printer info for hdr page. */
  2 header_pages bin,                       /* Number of header pages to print. */
  2 trailer_pages bin,                      /* Number of trailer pages to print. */
  2 evfu(MAX_EVFU_) bin,                    /* Storage for evfu after evfu_read. */
  2 stationery_feed_type bin,               /* Manual, Cut-sheet or Continuous. */
  2 despool_fatal_error label,              /* Fatal errors trap to here. */
  2 big_char char,                          /* Character used for big letters */
  2 old_pspass_now_RFU char(6),             /* PostScript password */
  2 log_type bin,                           /* Mode of log file */
  2 trailer_perf_lines bin,                 /* overprint lines on trl perfs */
  2 trailer_perf_char_flag bit(1) aligned,  /* if -PERF_CHAR option used */
  2 trailer_perf_char char,                 /* character for perf overprints */
  2 printronix_pi bit(1) aligned,           /* printronix paper instruction */
  2 swo_wait_time bin,                      /* Time to wait on EOF with -SWO */
  2 swo_times_to_check bin,                 /* No times to loop on -SWO & EOF */
  2 psi_increment bin(31),                  /* Word increment for %age printed */
  2 Map_table_ptr ptr,                      /* Pointer to current mapping table */
  2 sdb ptr,                                /* Pointer to despooler SIT context */
  2 aux_addr ptr;                           /* Pointer to auxiliary structure */
  2 env_name char(16) var,                  /* Environment name */
  2 evfu_ptr ptr,                           /* Pointer to evfu data */
  2 no_lf_after_ff bit(1) aligned,          /* Suppresses line feed after a form feed /*
  2 pspass char(32);                        /* PostScript password /*
```

Example of a Print Handler

This section lists an example of a print handler. It can serve as a template when you write a print handler of your own. The language used is PLP. Please note that the PRINTRONIX handler that is actually supplied as part of your Spooler subsystem may differ in some respects from the version listed here.

```
/* PRINTRONIX.PLP,SPOOLSRC>DESPOOL>HANDLERS,SPOOLER PROJECT,05/28/86
    PRINTRONIX printer handler
    Copyright (c) 1990, Prime Computer, Inc., Natick, Ma 01760 */

/* TITLE : PRINTRONIX - PRINTRONIX printer handler */

/* START-DESCRIPTION :
         s$init_                             assign device & initialize
         s$print_                            print a line of text
         s$validate_                         check for a valid ESCAPE sequence
         s$beep_                             make the printer beep
         s$input_                            accept input from device
         s$page_                             page throw
         s$status_                           read status from device
         s$start_doc_                        start of document
         s$end_doc_                          end of document
         s$large_print_                      print large format characters
         s$make_ready_                       ready printer for printing
         s$unready_                          put printer back to idle
         s$port_                             alter paper orientation
         s$font_                             alter font
         s$bin_                              alter paper feed mechanism

   END-DESCRIPTION */

/* START-CODE : */

PRINTRONIX:
   procedure (key, text, qcom, envcom, pagcom, code);

$Insert spoolsrc>insert>envcom.ins.plp

$Insert spoolsrc>insert>qcom.ins.plp

$Insert spoolsrc>insert>page_com.ins.plp

$Insert spoolsrc>insert>constants.ins.plp

$Insert spoolsrc>insert>device_info.ins.plp

$Insert spoolsrc>insert>cc_defns.ins.plp

$Insert spoolsrc>insert>s$keys.ins.plp

$Insert syscom>a$keys.ins.pl1

$Insert syscom>keys.ins.pl1

$Insert syscom>errd.ins.pl1

   dcl key bin                               /* action key */
   dcl text char(1024) var;                  /* text to process */
   dcl 1 qcom like qcom_based;               /* queue common */
   dcl 1 envcom like envcom_based;           /* environment common */
   dcl 1 pagcom like pagcom_based;           /* page common */
   dcl code bin;                             /* standard error code */

   dcl case$a entry(bin, char(*), bin);

%Replace p8_bel_ by '87'b4,                  /* Chars with Top bit on. PRIME 8. */
         p8_ff_ by '8C'b4,
         p8_lf_ by '8A'b4,
      p8_cr_ by '8D'b4,
      p8_esc_ by '8B'b4,
      p8_nul_ by '80'b4,
      p8_can_ by '98'b4;


%Replace nul_ by '00'b4;

   dcl pad_char char static internal;
   dcl large_print(7) char(132) var;
   dcl vs$ char(512) var;
   dcl Work_string char(1024) var;
   dcl max_big_letters bin;
   dcl i bin;
   dcl inst bin;
   dcl module char(32) var static init('PRINTRONIX');
   dcl 1 vs based,
         2 len bin,
         2 chars char(1024);
   dcl 1 ctl based,
         2 len bin,
         2 char1 bit(8) unaligned,
         2 char2 bit(8) unaligned,
         2 data(1022) bit(8) unaligned;
   dcl 1 ascii based,
         2 char bit(8) unaligned;
   dcl junk_string char(1024) var;

       code = 0;

       select(key);
       when(s$init_)
          do;
             envcom.printer_info(1) = '';
             envcom.printer_info(2) = '';
             envcom.banner_eject = '0'b;

/* check on whether we're to use the default pad character */

            if pagcom.pad_char_flag
            then                                   /* overwrite */
               pad_char = pagcom.pad_char;
            else
               pad_char = chr$(nul_);              /* the default */

/* check if an evfu format file is present */

            if envcom.evfu_sw & pagcom.lines_ppage <= max_evfu_
            then
               call addr(envcom) -> envcom_based.read_evfu(envcom);
            else
               do;
                  envcom.evfu(1) = 1;
                  do i = 2 to max_evfu_;
                     envcom.evfu(i) = 0;
                  end;
               end;

/* assign device */

            call addr(envcom) -> envcom_based.sp$dtr(s$asgn, null(), 0, 0, code);
            if code ^= 0
            then
               return;

/* down line load the evfu */

           if ^envcom.evfu_disabled_sw
           then
              select(envcom.device_type);
              when(sp$para_)
                 if envcom.printronix_pi
                 then
                    do;
                     inst = '20156'b3;

                     call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                               null (), 0, inst, code);
                      do i = 1 to pagcom.lines_ppage;

                         if envcom.evfu(i) = 0
                         then
                            inst = '20115'b3;
                         else
                            inst = '20077'b3 + envcom.evfu(i);
                         call addr(envcom) -> envcom_based.sp$dtr
                                   (s$putb, null(), 0, inst, code);
                      end;

                      inst = '20157'b3;
                      call addr(envcom) -> envcom_based.sp$dtr
                                (s$putb, null (), 0, inst, code);
                   end;
                else
                   do;                          /* not Paper_Instruction type printronix */
                      inst = '20036'b3;
                      call addr(envcom) -> envcom_based.sp$dtr
                                (s$putb, null (), 0, inst, code);
                      do i = 1 to pagcom.lines_ppage;
                         if envcom.evfu(i) = 0
                         then
                            inst = '20035'b3;
                         else
                            inst = '20017'b3 + envcom.evfu(i);
                         call addr(envcom) -> envcom_based.sp$dtr
                                   (s$putb, null(), 0, inst, code);
                      end;
                      inst = '20037'b3;
                      call addr(envcom) -> envcom_based.sp$dtr
                                (s$putb, null (), 0, inst, code);
                   end;

             otherwise                          /* serial */
                do;
                   vs$ = chr$('36'b3);
                   do i = 1 to pagcom.lines_ppage;
                      if envcom.evfu(i) = 0
                      then
                         vs$ = vs$ || chr$('35'b3);
                      else
                         do;
                            inst = envcom.evfu(i) + '17'b3;
                            vs$ = vs$ || chr$(inst);
                   end;
             end;
             vs$ = vs$ || chr$('37'b3);
             call addr(envcom) -> envcom_based.sp$dtr
                       (s$putb, addr (addr(vs$) -> vs.chars),
                         length(vs$), 0, code);
          end;
       end;
   end;
when(s$print_)
   call proc_print(text);
when(s$validate_)
   do;
   end;
when(s$beep_)
   do;
   end;
when(s$input_)
   text = '';
when(s$stop_print_)
   code = e$null;
when(s$status_)
   call addr(envcom) -> envcom_based.sp$dtr(s$lsta, null(), 0, 0, code);
when(s$page_)
   do;
      select(envcom.device_type);
      when(sp$para_)
       call addr(envcom) -> envcom_based.sp$dtr(s$putb, null(), 0,'20014'b3, code);
      otherwise
         do;
            Work_string = CHR$(p8_ff_);
            call MAP_CC_AND_OUTPUT(Work_string, Code);
         end;
      end;

      call print(text);
   end;
when(s$start_job_)
 call addr(envcom) -> envcom_based.sp$dtr(s$strt, null(), 0, 0, code);
when(s$start_doc_)
   do;
   end;
when(s$end_doc_)
   do;
   end;
when(s$end_job_)
 call addr(envcom) -> envcom_based.sp$dtr(s$endt, null(), 0, 0, code);
when(s$large_print_)
   do;
      max_big_letters = divide(pagcom.chars_pline, 6, 15);
      if max_big_letters > 22
      then
         max_big_letters = 22;

      if length(text) > max_big_letters
      then
         text = substr(text, 1, max_big_letters);

      call skip_lines(2);
      call addr(envcom) -> envcom_based.big_letters(text,
                large_print, envcom.big_char);
            do i = 1 to 7;
               call print((large_print(i)));
            end;

            call skip_lines(2);
         end;
      when(s$make_ready_)                         /* ready printer for printing */
         do;
            call addr(envcom) -> envcom_based.sp$dtr(s$mrdy, null(), 0, 0, code);


/* down line load the evfu */

           if ^envcom.evfu_disabled_sw
           then
              select(envcom.device_type);
              when(sp$para_)
                 if envcom.printronix_pi
                 then
                    do;
                     inst = '20156'b3;

                     call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                               null (), 0, inst, code);
                      do i = 1 to pagcom.lines_ppage;

                         if envcom.evfu(i) = 0
                         then
                            inst = '20115'b3;
                         else
                            inst = '20077'b3 + envcom.evfu(i);
                         call addr(envcom) -> envcom_based.sp$dtr
                                   (s$putb, null(), 0, inst, code);
                      end;

                      inst = '20157'b3;
                      call addr(envcom) -> envcom_based.sp$dtr
                                (s$putb, null (), 0, inst, code);
                   end;
                else
                   do;                            /* not Paper_Instruction type printronix */
                      inst = '20036'b3;
                      call addr(envcom) -> envcom_based.sp$dtr
                                (s$putb, null (), 0, inst, code);
                      do i = 1 to pagcom.lines_ppage;
                         if envcom.evfu(i) = 0
                         then
                            inst = '20035'b3;
                         else
                            inst = '20017'b3 + envcom.evfu(i);
                         call addr(envcom) -> envcom_based.sp$dtr
                                   (s$putb, null(), 0, inst, code);
                      end;
                      inst = '20037'b3;
                      call addr(envcom) -> envcom_based.sp$dtr
                                (s$putb, null (), 0, inst, code);
                   end;

             otherwise                            /* serial */
                do;
                   vs$ = chr$('36'b3);
                   do i = 1 to pagcom.lines_ppage;
```

```
                      if envcom.evfu(i) = 0
                      then
                         vs$ = vs$ || chr$('35'b3);
                      else
                         do;
                            inst = envcom.evfu(i) + '17'b3;
                            vs$ = vs$ || chr$(inst);
                         end;
                   end;
                   vs$ = vs$ || chr$('37'b3);
                   call addr(envcom) -> envcom_based.sp$dtr
                             (s$putb, addr (addr(vs$) -> vs.chars),
                               length(vs$), 0, code);
                end;
             end;


/* Example of use of AUXILIARY parameters.
   If there is anything held in element one of the auxiliary structure
   then send it down to the printer.      This code has been commented
   out! It serves only as an example of how to access auxiliary
   parameters

            if envcom.aux_addr -> aux_based.auxiliary(1) ^= ''
            then
               call addr(envcom) ->envcom_based.sp$dtr(s$putb,
                         envcom.aux_addr -> aux_based.auxiliary(1),
                         length(envcom.aux_addr -> aux_based.auxiliary(1)),
                         '40000'b3, code);

   End of example.  Note how the address of the auxiliary structure
   is stored in envcom.aux_addr */

         end;
      when(s$unready_)                         /* put printer back to idle */
         do;
            call addr(envcom) -> envcom_based.sp$dtr(s$urdy, null(), 0, 0, code);
         end;

      when(s$port_)                            /* paper orientation */
         do;

/* this handlers doesn't support paper orientations */

            code = e$nsuc;
         end;
      when(s$font_)/* font */
         do;

/* this handlers does'nt support fonts */

            code = e$nsuc;
         end;

      when(s$bin_)                             /* paper_bin */
        do;

/* this handler doesn't support paper bins */

            code = e$nsuc;
         end;
      otherwise                                 /* unrecognised key */

         code = E$BKEY;

      end;

      return;


/* ============================== PROC_PRINT */

PROC_PRINT:
      procedure (text);

      dcl text char(1024) var;
      dcl channel bin;

        if length(text) = 0
        then
           text = ' ';

        select(addr(text) -> ctl.char1 & '7F'b4);

        when('02'b4)                            /* found a control record */

          call control(text);

        when('03'b4)                            /* found a skip to EVFU channel */

          do;
             if length(text) = 1
             then
                channel = 1;                    /* set default */
             else
                channel = addr(text) -> ctl.char2 & '7F'b4;

             call skip_channel(channel);

             if length(text) > 2
             then
                text = substr(text, 3);
             else
                text = '';
          end;
       end;

       call print(text);

    end;

/* ================================= PRINT */

PRINT:
      procedure (text);

      dcl text char(1024) var;

      dcl code bin;

      dcl required bit(1);

      dcl current_position bin,
          output_string char(1024) var,
          last_status bin,
          status bin;

      dcl sp$map entry(ptr, char(1024) var, bin, char(1024) var, bin, bin),
                      ioa$ entry options(variable);

         if length(text) = 0
         then
            return;

         if envcom.uc_sw                                /* convert to uppercase if necessary */
         then
            call case$a(A$FUPP, addr(text) -> vs.chars, length(text));

/* print the data */

         select(envcom.device_type);
         when(sp$para_)
            do;

               current_position = 0;
               status = 0;
               required = '1'b;
               do while(required);

                  last_status = status;
                  status = 0;

                  /* map the text ready for output. */

                  call sp$map(envcom.map_table_ptr, text,
                       current_position,output_string,
                       last_status, status);

                  if status <= -1000
                  then
                     do;                           /* somebody has defined a file with 1:N */
                                                   /* substitutions in it and it has */
                                                   /* overflowed the buffer */
                        call ioa$('%v%.', -1,
                        'substitution has caused buffer overflow in
                        output.'); call ioa$('%v%.', -1,
                                   'Output may not print correctly.');
                     end;

                  else
                     if status > 0
                     then
                        do;
                           code = status;
                           return;
                        end;

/* make sure its a whole number of words */

                 if mod(length(output_string), 2) ^= 0
                 then
                    output_string = output_string || pad_char;

                 call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                           addr(addr(output_string) -> vs.chars),
                           length  (output_string), '40000'b3, code);
                                               /* output the line but do not send CR/LF */

                 if (status < 0)               /* multi-pass mapping required */
                 then
                    do;                        /* Check if need to force printposition back
                                                  to first column */
                      if status > -1000
                      then
                         do;

/* No need as each line will overprint unless new line forced */

                         end;
                    end;

                 else                          /* else must be zero */
                    required = '0'b;

              end;

/* send appropriate line terminator */

              if pagcom.append_lf
              then
                 call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                           null(), 0, '20012'b3, code);

            end;

         otherwise
            do;

/* append appropriate line terminator if not already present */

               if substr(text, length(text), 1) ^= chr$(p8_cr_)
               then
                  text = text || chr$(p8_cr_);

/* --- map the output and then print it. */

               current_position = 0;
               status = 0;
               required = '1'b;
               do while(required);

                  last_status = status;
                  status = 0;
                  call sp$map(envcom.map_table_ptr, text,
                            current_position, output_string,
                            last_status, status);

                  if status > 0
                  then
                     do;
                        code = status;
                        return;
                     end;

                  call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                            addr(addr
                            (output_string) -> vs.chars), addr
                            (output_string) -> vs.len, 0, code);

                  if status = 0 | code ^= 0
                  then
                     required = '0'b;
                  else
                     if status > -1000
                     then
                        do;                    /* force print position
                                                  back to first column */
                           work_string = chr$(p8_cr_);
                           call map_cc_and_output(work_string, code);

                        end;

                end;

                if envcom.crlf_sw & pagcom.append_lf
                then
                   do;
                      work_string = chr$(p8_lf_);
                      call map_cc_and_output(work_string, code);
                   end;

            end;

         end;

/* increment line count */

         if pagcom.append_lf
         then
            do;
               pagcom.virt_line_count = pagcom.virt_line_count + 1;
               if pagcom.print_it | pagcom.override_from_to
               then                            /* only increment if printing */
                  pagcom.line_count = pagcom.line_count + 1;
            end;

      end;

/* ================================== CONTROL */

CONTROL:
      procedure (text);

      dcl text char(1024) var;

      dcl lines bin;

        if length(text) = 1                    /* ignore invalid control record */
        then
           return;

        select(addr(text) -> ctl.char2 & '7F'b4);
        when(set_lpi_, set_cpi_, set_lpp_, set_cpl_, set_tab_int_,set_tabs_)
           if length(text) > 3
           then
              text = substr(text, 4);
           else
              text = '';
        when(clear_tabs_, set_uc_, clear_uc_)
```

```
           if length(text) > 2
           then
              text = substr(text, 3);
           else
              text = '';
           when(set_channel_)
              do;
                 i = addr(text) -> ctl.data(2) & '7F'b4;
                 if i <= max_evfu_
                 then
                    envcom.evfu(i) = addr(text) -> ctl.data(1) & '7F'b4;
                 if length(text) > 4
                 then
                    text = substr(text, 5);
                 else
                    text = '';
              end;
           when(clear_channel_)
              do;
                 envcom.evfu(1) = 1;
                 do i = 2 to max_evfu_;
                    envcom.evfu(i) = 0;
                 end;
                 if length(text) > 2
                 then
                    text = substr(text, 3);
                 else
                    text = '';
              end;
           when(skip_lines_)                   /* skip n lines */
              do;
                 if length(text) = 2           /* no default */
                 then
                    text = '';
                 else
                    do;
                       lines = addr(text) -> ctl.data(1) & '7F'b4;
                       call skip_lines(lines);
                       if length(text) > 3
                       then
                          text = substr(text, 4);
                       else
                          text = '';
                    end;
              end;

           end;

        end;

/* ================================ SKIP_LINES */

SKIP_LINES:
     procedure (lines);

     dcl lines bin;

     dcl n15 bin;
     dcl r15 bin;

       if lines = 0
       then
```

```
          return;

       if lines > 256
       then
          lines = 256;

       select(envcom.device_type);
       when(sp$para_)
          if ^envcom.evfu_disabled_sw
          then
             do;
                n15 = divide(lines, 15, 15);   /* max slew is 15 lines */
                r15 = mod(lines, 15);

                do while(n15 > 0);
                   call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                             null(), 0; '20137'b3, code);
                   n15 = n15 - 1;
                end;

                if r15 > 0
                then
                   do;
                      inst = '20120'b3 + r15;
                      call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                                null (), 0, inst, code);
                   end;
             end;
          else
             do i = 1 to lines;
                call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                          null(), 0, '20012'b3, code);
             end;
       otherwise                               /* serial */
          do;
             work_string = '';
             do i = 1 to lines;
                work_string = work_string || chr$(p8_lf_);
             end;

             call map_cc_and_output(work_string, code);
          end;
       end;

       pagcom.virt_line_count = pagcom.virt_line_count + lines;

    end;

/* =============================== SKIP_CHANNEL */

SKIP_CHANNEL:
      procedure (channel);

      dcl channel bin;

      dcl inst bin;
      dcl i bin;
      dcl lines bin;
      dcl found bin;
      dcl ff char(2);
      dcl ch char(2);
      dcl work_string char(1024) var;
      dcl output_string char(1024) var;

      dcl sp$map entry(ptr, char(1024) var, bin, char(1024) var,
                            bin, bin);

/* find out where a skip to this channel will take us
   so that line count may be kept up to date */

         found = 0;

         do i = pagcom.virt_line_count to max_evfu_ while(found = 0);
            if envcom.evfu(i) = channel
            then
               found = i;
         end;

         do i = 1 to pagcom.virt_line_count while(found = 0);
            if envcom.evfu(i) = channel
            then
               found = i;
         end;

/* if channel does not exist just ignore the skip */

         if found = 0
         then
            return;

         if ^envcom.evfu_disabled_sw
         then
            select(envcom.device_type);
            when(sp$para_)                 /* parallel */
               if envcom.printronix_pi
               then
                  do;
                     inst = '20077'b3 + channel;
                     call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                               null(), 0, inst, code);
                  end;
               else
                  do;                      /* not Paper_Instruction type printronix */
                     inst = '20017'b3 + channel;
                     call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                               null(), 0, inst, code);
                  end;
            otherwise                      /* serial */
               do;
                  inst = '17'b3 + channel;
                  ch = chr$(inst) || chr$(nul_);
                  call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                            addr(ch), 1, 0, code);
                  pagcom.virt_line_count = found;
               end;
            end;
         else
            do;
               if found < pagcom.virt_line_count
               then
                  do;
                     select(envcom.device_type);
                     when(sp$para_)
                        call addr(envcom) -> envcom_based.sp$dtr(s$putb,
                                    null (), 0, '20014'b3, code);
                     otherwise
                        do;
                           work_string = chr$(p8_ff_);
                           call map_cc_and_output(work_string, code);
                        end;
                     end;

                     call addr(envcom) -> envcom_based.page_throw;
                  end;

               lines = found - pagcom.virt_line_count - 1;

               call skip_lines(lines);

         end;

      end;

/* ======================== MAP_CC_AND_OUTPUT */

MAP_CC_AND_OUTPUT:
      procedure (in_string, code);

/* this routine maps and outputs control code sequences to the
   printer. It only works for serial lines. */

/* it DOES NOT cope with substitutions which require the printing
   of more than one line (Overprinting substitutions) */

    dcl in_string char(1024) var,
        temp_string char(1024) var,
        max bin,
        i bin,
        length_input bin,
        output_string char(1024) var,
        code bin,
        status bin,
        last_status bin,
        current_position bin;

    dcl sp$map entry(ptr, char(1024) var, bin, char(1024) var,
                   bin, bin);

       length_input = length(in_string);
       max = divide(length_input - 1, 32, 15);

       do i = 0 to max;
          if (i + 1) * 32 < length_input
          then
             temp_string = substr(in_string, i * 32 + 1, 32);
          else
             temp_string = substr(in_string, i * 32 + 1, length_input);

          current_position = 0;
          status = 0;
          last_status = 0;

          /* map the text ready for output. */

          call sp$map(envcom.map_table_ptr, temp_string,
                      current_position, output_string, last_status, status);
          if (status ^= 0)                     /* if not fully mapped */
          then
             do;
                code = status;
                return;
             end;

          call addr(envcom) -> envcom_based.sp$dtr(s$putb, addr(addr
                    (output_string) -> vs.chars),
                      length(output_string), 0, code);
                                               /* output the line */

       end;
    end;

/*  ==================================== CHR$ */

CHR$:
     procedure (i) returns(char);

     dcl i bin;

     dcl 1 ascii based,
            2 c1 bit(8) unaligned,
            2 c2 char(1) unaligned;

        return(addr(i) -> ascii.c2);

     end;

   end;

/* END-CODE */
```

C

SPOOLER SUBSYSTEM MESSAGES

This appendix lists the messages and the message code that the Spooler subsystem displays. For clarity, Severity, Localizable Subsystem id, Non Localizable Subsystem id, and Module id have been omitted. An explanation is given where more detail is helpful, and the steps that you must take to rectify an error are explained where relevant.

The messages, including a message code listed in the following section, Spooler Subsystem Messages, will appear on your terminal in the following format:

```
Error from spooler (Spool-49)
Invalid command line option: -OPT
```

The message consists of the following components:

Severity	:	Error
Message Body	:	Invalid command line option : -OPT
Localizable Subsystem id	:	Spooler
Non localizable subsystem id	:	spool
Message code	:	49

The message code is used to identify the message easily.

Explanations are not given for DPTX error messages.

When you use the PROP and SPOOL commands it is possible that you will obtain a message that is not listed in this appendix. This means that there is an error in a PRIMOS subsystem that is used by the Spooler subsystem.

You can obtain Prime's support package for Public Data Networks (PDN) separately. For convenience, PDN messages are listed separately at the end of this appendix.

Spooler Subsystem Messages

`-ALIAS not allowed with pre-rev. 21 queue`

Message code 7. You have used the -ALIAS option of the SPOOL command and have also used -DISK to specify a pre-Rev. 21.0 spool queue. Issue another SPOOL command, either without the -ALIAS option, or directed to a Rev. 21.0, or post Rev. 21.0, queue.

`** Aborted **`

The previous print job was aborted, in response to a PROP command with the -ABORT option. No action required.

`Access to pre-rev. 21 style queue on disk aa failed with error nn`

Message code 22. The SPOOL command was unable to access the specified disk. The fault is not in the Spooler subsystem but in the disk access. Appropriate action depends on reason for failure.

`Access to spool queue failed`

Message code 6. The SPOOL command was not able to access the spool queue. Check that the queue exists, is visible from your node, and has the correct ACL.

`Address not numeric`

Message code 127. Mistake in environment file. Correct the environment file.

`Address too long`

Message code 128. Mistake in environment file. Correct the environment file or use a different environment.

`Allocating semaphore`

Message code 2. Status message from the despooler phantom. No action required.

`-ALTernate has been used on an earlier line.\R\L`

Message code 191. More than one MAP directive has the -ALT option to it. Check MAP directives.

`Argument needed with -SET_FONT`

Message code 58. You have omitted fontname. Include fontname.

`Argument needed with -SET_PAPER_BIN`

Message code 59. You have omitted n or MANUAL. Include n or MANUAL.

`Attaching to SPOOL* directory`

Message code 8. A PRIMOS error message has occurred while attaching to the SPOOL* directory. See your system administrator.

`Attaching to top-level SPOOL* directory`

Message code 3. Status message. No action required.

`Attaching to SPOOL*>LOG sub directory`

Message code 9. Status message. No action required.

`Attribute too long (max 16 chars)`

Message code 149. Mistake in the environment file. Correct the environment file or use a different environment.

`** Backed Up **`

The backup instruction issued with the -BACKUP option of the PROP command has been successfully obeyed. No action required.

`Cannot alter RWLOCK to UPDT for -SPOOL_WHILE_OPEN`

Message code 77. The spooler cannot change the RWLOCK to UPDT. Change the RWLOCK on the file to UPDT.

`-COMPRESS is not available in this release`

Message code 156. You have tried to use the -COMPRESS option of the PROP command. This option was removed at Rev. 21.0. Make all changes to environment files with a text editor.

`-CREATE is not available in this release`

Message code 157. You have tried to create a printer environment with the -CREATE option of the PROP command. This option was removed at Rev. 21.0. See Chapter 3, Defining Printer Environments, for the correct procedure.

`Cannot attach to queue directory on specified disk`

Message code 26. The SPOOL command could not attach to the directory that contains the spool queue on the specified disk. Check the ACL on the directory that it needs to attach to.

`Cannot attach to SPOOL*>RESTRICTED_USERS directory`

Message code 10. The spooler cannot attach to the directory. Check to see if the directory is there.

`Cannot cancel request nn as the despooler already has a request waiting`

Message code 123. You have used the -CANCEL option of the SPOOL command to cancel a job that has already begun to print. In this particular case the despooler phantom already has a request pending (for example to stop at the end of the current job). It is only able to have one request pending and so cannot handle your request. Use the -DROP option of the PROP command instead. If you are not a privileged user, contact the operator or system administrator.

`Cannot cancel request nn as it is being printed remotely`

Message code 124. You can only cancel print jobs on the local node of a network. Either contact the operator of the remote system, or log in on the remote system and cancel the print job yourself.

`Cannot find handler`

The despooler initialization routine could not find the print handler specified in the environment file. Check the environment file to make sure that the print handler is specified correctly. The problem could be a simple spelling mistake.

If the environment file is correct, check that the print handler exists. See Chapter 2, Setting Up the Spooler Subsystem, for a list of the standard handlers that Prime supplies. If the print handler is not in the standard library, ensure that you have provided a suitable nonstandard handler, and that the search rules in SPOOL* >ENTRY$.SR are modified so that it can be found.

`Cannot modify request as it includes information not recognized by this rev.`

Message code 53. You have used the -MODIFY option of the SPOOL command to change the details of a print request that is not in Rev. 21.0 format, or later. Cancel the request and issue a new SPOOL command.

`Cannot modify request as it is in rev. 18 format`

Message code 61. You have used the -MODIFY option of the SPOOL command to change the details of a print request that is not in Rev. 21.0 format,or later. Cancel the request and issue a new SPOOL command.

`Cannot open attributes file`

Message code 29. The SPOOL command tries to check the attributes in a SPOOL command against those in an attributes file. The file that cannot be opened could be user specific, project specific, or the default file. Find out which file is locked, and why.

`Cannot open file to print`

Message code 21. Either the directory specified does not exist, or the file does not exist, or a failure other than NO_RIGHTS (E$NRIT) has occurred when trying to change the RWLOCK on the file, or the file cannot be opened for reading. Check that the file exists, is not locked, and that the ACL in force allows the despooler phantom to access the file.

`Cannot open queue file`

Message code 30. The SPOOL command cannot open the spool queue to put a new request in it. Find out which process has a lock on the file and why.

`Changing name of log file`

Message code 11. The log file name is being changed as specified in the LOG command of the environment file. No action required.

`Check New Page`

Operator instruction. Check the page at the printer.

`Check Paper Alignment`

Operator instruction. Check the paper alignment at the printer.

`Closing semaphore`

Message code 155. Status message from the PROP command. For details of semaphores see the Prime 50 series Technical Summary. No action required.

`Command line contains invalid characters`

Message code 42. You have issued a SPOOL command with non alphanumeric characters in it. Possibly you have pressed the CTRL key inadvertently. Issue the correct SPOOL command.

`Command not available. System date and time not set.`

Message code 15. The system date and time were not set at system cold start. Until they are set the SPOOL command cannot be used. Set the system date and time and reissue the PROP -COLDSTART command. Ensure that the system date and time are correctly set at subsequent system cold starts.

`** Completed **`

Confirmation that the specified print request is completed. This message is only given if you used the -NOTIFY option of the SPOOL command. No action required.

`Conflicting options`

Message code 114. Mistake in the environment file. Correct the environment file or use a different environment.

`Creating SEM file`

Message code 4. Status message. As part of despooler initialization a SEM file is created to allow communication with the semaphore where the despooler phantom is queued when it has no work to perform. For details of semaphores see the Prime 50 series Technical Summary. No action required.

`Default already specified`

Message code 16. You have allocated -DEFAULT twice when writing SET statements in the environment file. Reconsider your SET statements.

`-DEFault has been used on an earlier line.\R\`

Message code 192. More than one MAP directive has the -DEF option to it. Check the MAP directives.

`-DELETE is not available in this release`

Message code 159. You have tried to delete an environment with the -DELETE option of the PROP command. This option was removed at Rev. 21.0. See Chapter 3, Defining Printer Environments, for the correct procedure.

`-DISK and -ON may not be used together`

Message code 17. You have given a SPOOL command that includes both the -DISK option and the -ON option. You should use -DISK to specify a pre-Rev. 21 spool queue on that disk, and -ON to specify the queue on a Rev. 21.0 node. Either decide which of the options you intended to use, or give separate commands with -ON and -DISK to put the request in two different spool queues.

`DPTX: Inputting from Printer`

DPTX status information. No action required.

`DPTX: Outputting to Printer`

DPTX status information. No action required.

`Despooler already has an outstanding request`

Message code 166. You have issued a PROP command to tell a despooler phantom to take some action, but the despooler phantom has already been given its next instruction. Wait until the despooler carries out the pending instruction, and re-assess the situation.

`Despooler not currently printing`

Message code 161. You have issued the PROP command with the -DROP option, but the despooler phantom was not printing a file. No action required.

`Despooler not currently hanging`

Message code 160. You have given a PROP command with the -CONTINUE action, but the despooler phantom was not hanging. No action required.

`Device not recognised`

Message code 118. Mistake in the environment file. Correct the environment file.

`Device not responding`

Reports that a printer running over TCP/IP has gone off line, or run out of paper. Appears in the log file.

`Disk aa not known`

Message code 18. You have used the -DISK option of the SPOOL command to specify a disk, where *aa* is the disk name, but the disk is not known to your node of the network. Make sure that the partition is visible from your node of the network.

`** Dropped **`

Confirmation that a printing job has been dropped, in response to a PROP command. No action required.

`Duplicated attribute name`

Message code 19. You have given the same attribute more than once on SPOOL command line. Issue the correct SPOOL command.

`-ELEMENT must be specified`

Message code 188. You have not specified -ELEMENT when defining an AUXILIARY statement in the environment file. Include -ELEMENT.

`EVFU Channel 1 is for top of form only`

Message code 178. Mistake found in EVFU file. Correct the EVFU file.

`EVFU channel 1 must be top of form`

Message code 181. Mistake found in EVFU file. Correct the EVFU file.

`Environment already active`

Message code 109. You have tried to use the PROP command to start an environment that is already active. No action required, unless the command you gave was not what you intended.

`Environment name missing`

Message code 167. You have omitted an environment name from a PROP command. Issue the correct command.

`Environment name not required`

Message code 164. You have given an environment name in a PROP command where it was not needed. Issue the correct command.

`Environment name too long`

Message code 165. Mistake in PROP command. Issue the correct command. If the name you gave really is the name under which the environment file is saved than you must rename the environment file with a shorter name. Environment names can be a maximum of 16 characters.

`Environment not active`

Message code 163. You have used the PROP command to stop or hang an environment that is not active, or to instruct an inactive environment to drop or abort a print job. This is most likely due to a typing mistake. Issue the command that you intended.

`Error reading file to be spooled`

Message code 12. This is a disk handling error. Reissue the SPOOL command. If the error recurs investigate the disk drive and controller.

`Error reading SWO_VALID_USERS file`

Message code 97. An error has occurred in reading the file. Check with your system administrator.

`Error writing data file copy`

Message code 13. This is a disk handling error. Reissue the SPOOL command. If the error recurs investigate the disk drive and controller.

`Spooler database has not been shared`

Message code 171. The correct procedures were not followed when the Spooler subsystem was started. Perform the startup procedure correctly, as described in Chapter 2, Setting Up the Spooler Subsystem.

`Failed to add entry`

Message code 24. The SPOOL command software could not add an entry to the spool queue. The reason is given in the rest of the message. Appropriate action depends on the reason for failure.

`Failed to add entry to <ldev>SPOOL_DATA* - n`

Message code 25. The SPOOL command software could not copy the file to be printed to the SPOOL_DATA* directory on the named partition. The reason is given in the rest of the message. Appropriate action depends on the reason for failure.

`Failed to delete request`

Message code 27. The SPOOL command software could not remove a request from the spool queue. The reason is given in the rest of the message. Appropriate action depends on the reason for failure. Note that you can only cancel your own requests, unless you are a privileged user.

`Failed to modify request`

Message code 28. The SPOOL command software could not modify a request in the spool queue. The reason is given in the rest of the message. Appropriate action depends on the reason for failure. Note that you can only modify your own requests, unless you are a privileged user.

`Failed to open PRT file`

Message code 23. The Spooler subsystem could not open a PRT file for your print request. The reason is given in the rest of the message. Appropriate action depends on the reason for failure.

`Failed to read queue file header`

Message code 31. The Spooler subsystem was unable to read the header data in the queue file. Reissue the command that led to the error. If the error recurs, the spool queue file might be corrupted. This could have resulted from a disk error.

Follow your installations procedures for recovering from data loss. Restore the Spooler directories to their most recent known state, but retain copies of files that were recently added to the SPOOL_DATA* directories, in case the originals have been deleted from the file system. Notify users of the event, in case they need to reissue some print requests.

`Failed to read queue file`

Message code 32. The Spooler subsystem could not read the spool queue file. The reason is given in the rest of the message. Appropriate action depends on the reason for failure.

`Failed to read request`

Message code 33. The Spooler subsystem could not read an entry in the spool queue. The reason is given in the rest of the message. Appropriate action depends on the reason for failure.

`Failed to write queue entry`

Message code 34. The SPOOL command software could not enter a request in the spool queue. The reason is given in the rest of the message. Appropriate actions depends on the reason for failure.

`Failed to write queue file header`

Message code 35. The Spooler subsystem could not update the header of the spool queue file. The reason is given in the rest of the message. Appropriate action depends on the reason for failure.

`File is being printed`

Message code 93. You have used the -MODIFY option of the SPOOL command to alter the options or attributes of a print requests, but the changes cannot be made because the file is already printing. If the original command will produce an unacceptable printout, cancel the print job, and issue a new SPOOL command with the correct options and attributes.

`Filename only please, no wildcard`

Message code 36. You have used wildcards to name a map in the environment file. Use explicit filenames only.

`Fontname required with -SET_FONT`

Message code 57. You have omitted fontname. Include the fontname.

`-FROM not allowed with pre-rev. 21 queue`

Message code 72. You are trying to use the -DISK and -FROM options together. See your system administrator.

`FROM value must not be greater than TO value`

Message code 38. You have given a FROM value which is greater than the TO value. Check FROM and TO values.

`** Hanging **`

Confirmation that PROP command has been obeyed. No action required.

`-HDR not allowed with pre-rev. 21 queue`

Message code 48. You are trying to use the -DISK and -HDR options together. See your system administrator.

`Illegal channel number in EVFU file`

Message code 179. Mistake in EVFU file. Correct the EVFU file.

`Illegal line number in EVFU file`

Message code 180. Mistake in EVFU file. Correct the EVFU file.

`Incompatible attributes : aaa, aaa`

Message code 39. You have given a SPOOL command with two attributes that describe the same feature. Issue the correct SPOOL command.

`Incompatible options`

Message code 40. You have given a SPOOL command with two or more options that do not make sense together. Issue the correct SPOOL command.

`Initialized`

Confirmation that a print handler is successfully initialized. No action required.

`Initializing printer handler`

Initialization of the despooler's print handler has started. No action required.

`Insufficient access rights\R\L`

Message code 185. The user is not priviledged to use the failed PROP option. See your system administrator.

`Invalid buffer returned from node aa`

Message code 47. Network problem. Check the status of the remote node. *aa* is the node name. If the node is Up, reissue the command.

`Invalid character in attribute name : aaa`

Message code 41. An attribute in your SPOOL command contains non alphanumeric characters. *aaa* is the attribute that you entered incorrectly. If it appears to be correct you may have accidentally entered a control character. Issue the correct SPOOL command.

`Invalid command line. Type SPOOL -HELP for assistance`

Message code 43. Use SPOOL -HELP or refer to Appendix A, Summary of Spool and Prop Commands.

`Invalid command line option : -aaa`

Message code 49. You have given an incorrect option in a SPOOL command. *-aaa* is the incorrect option. Check for a spelling mistake. If the option appears to be correct you may have accidentally entered a control character. Refer to Appendix A, Summary of Spool and Prop Commands, for details of the command options that you can use.

`Invalid device type`

An invalid device type is specified in the environment file. Check the environment file. Either specify one of the available devices, or provide a print handler for the new device type that you want to use.

`Invalid Environment File`

Message code 45. Mistake in the environment file. Correct the environment file.

`Invalid -FCTY data`

Message code 129. Mistake in the environment file. Correct the environment file.

`Invalid language name`

Message code 190. The call to SI$VLA has failed whilst trying to validate a language name. You have used an unsupported language name. Issue command correctly.

`Invalid number of header pages (must be 0/1/2)`

Message code 15. Mistake in the environment file. Correct the environment file.

`Invalid page width`

Message code 152. Mistake in the environment file. Correct the environment file.

`Invalid parameter (must be ON/OFF)`

Message code 120. Mistake in the environment file. Correct the environment file.

`Invalid parameter (must be ON/OFF/SAVE)`

Message code 111. Mistake in the environment file. Correct the environment file.

`Invalid -PRID data`

Message code 130. Mistake in the environment file. Correct the environment file.

`Invalid records increment, must be > 0`

Message code 115. Mistake in the environment file. Correct the environment file.

`Invalid second count for Spool While Open Wait Time (must be 1-600)`

Message code 117. Mistake in the environment file. Correct the environment file.

`Invalid -SET_FONT`

Message code 46. You have not given a valid fontname. Recheck the command.

`Invalid -SET_PAPER_BIN`

Message code 50. You have specified a bin number outside the range 1 through 9, or not typed the string MANUAL correctly. Recheck the command.

`Invalid Spool While Open Times To Check (must be 0-32767)`

Message code 116. Mistake in the environment file. Correct the environment file.

`Invalid stationery feed type`

Message code 134. Mistake in the environment file. Correct the environment file.

`Invalid time in -DEFER`

Message code 44. You have given a SPOOL command with an incorrect defer time. The time must be in the range 0000 through 2359, though a number of formats are permitted. Reissue the SPOOL command with the time specified correctly. See Appendix A, Summary of Spool and Prop Commands, for details.

`-LINE option missing`

Message code 148. Error in the environment file, -LINE is missing from the ASYNC command. Correct the environment file.

`-LIST -ALL specified with -DISK or -ON`

Message code 52. You have given a SPOOL command to view the spool queue, but have used an incorrect combination of options. SPOOL -LIST -ALL shows all the spool queues so neither -DISK nor -ON is needed. Issue the correct command to display the queues you wish to see.

`Looking for a further SPOOL_DATA* partition`

The SPOOL command copies files to be printed into a SPOOL_DATA* directory. It has found that the SPOOL_DATA* directory that was previously in use is full and is now looking for another SPOOL_DATA* directory on a different partition. No action required.

`Map not defined`

Message code 54. Despooler log file message. You have called a MAP which does not exist in the environment file. Reconsider required MAP.

`-MODIFY is not available in this release`

Message code 169. You have tried to use the -MODIFY option of the PROP command to alter an environment. This option was removed at Rev. 21.0. See Chapter 3, Defining Printer Environments, for the correct procedure.

`Missing Command Line Option`

Message code 168. You have issued a PROP command with no command–line option. Issue the correct command.

`Missing map name or filename`

Message code 60. You have written a MAP without giving a filename. Recheck map in the environment file.

`Missing parameter`

Message code 112. Mistake in the environment file. Correct the environment file.

`Missing qualifier for option -$`

Message code 122. Missing qualifier for option –option_name. The option expected an argument, but none were given. Issue the correct command.

`More than one file specified`

Message code 101. You have given more than one pathname in a SPOOL command. This message could result from a typing mistake. If you miss the hyphen off an option then the remainder of the option could be interpreted as a file name. If you do wish to print several files then either issue a separate SPOOL command for each, or use the PRIMOS iteration facility.

`More than one print format specified`

Message code 63. You have specified more than one print format in the same SPOOL command, for example, -COBOL and -NOFORMAT. Reissue the command with only one formatting option.

`Multiple instances of SET_LANDSCAPE`

Message code 64. You have given more than one SET_LANDSCAPE statement in an environment file. Correct the environment file.

`Multiple instances of SET_PORTRAIT`

Message code 67. You have given more than one SET_PORTRAIT statement in an environment file. Correct the environment file.

`Multiple occurrences of -DISK not allowed`

Message code 62. You have given more than one -DISK option in the same SPOOL command. If you do want to enter the request in several spool queues, issue a separate SPOOL command for each one.

`Multiple occurrences of -ON not allowed`

Message code 65. You have given more than one -ON option in the same SPOOL command. If you do want to enter the request in several spool queues, issue a separate SPOOL command for each one.

`-<OOO> must be followed by a decimal value`

Message code 83. You have not given a decimal value to an option. Check your SPOOL command line.

`-NO_COPY not allowed with pre-rev. 21 queue`

Message code 69. You have used -NO_COPY and -DISK in the same SPOOL command. If you want the request to go into the pre-Rev. 21.0 spool queue you cannot use the -NO_COPY option. If you want to use -NO_COPY then you must direct the request to a Rev. 21.0 spool queue.

`-NPH not allowed with pre-rev. 21 queue`

Message code 8. You are trying to use the -DISK and -NPH options together. See your system administrator.

`No active environments`

Response to PROP -STATUS when no environments are active. No action required.

`No despooler environments`

Response to PROP -STATUS -ALL when there are no environment files in the SPOOL* directory on your system. No action required.

`No file name given`

Message code 70. You have omitted the file name from a SPOOL command. Issue the SPOOL command with the file name in place.

`No PDN support`

Message code 132. Your system does not include Public Data Network support software. Check with your administrator.

`No queue entries found to display`

Response to a SPOOL -LIST command. This does not necessarily mean that the queue is empty. If you are not a privileged user it means that none of your print requests are in the queue. Appropriate action depends on reason for viewing the spool queue.

`No space in environment table`

Message code 110. The Spooler subsystem keeps data for all active despooler phantoms in an environment table. You have tried to start a new despooler phantom, but there is insufficient space in the environment table to store data for it.

If there is no redundant data in the environment table you could still get this message if there are many active despooler phantoms. There is no precise maximum, but the environment table should accommodate more than 100 active despooler phantoms, so the theoretical limit should never be approached in practice.

If a despooler phantom is not shut down correctly, for example if it logs out abnormally, its data is not always successfullyremoved from the environment table. Shut down all despooler phantoms that are not required, with the -STOP option of the PROP command.

`Node name not required`

Message code 154. Mistake in the environment file. Correct the environment file or use a different environment.

`Non-candidate for despooling`

Refers to an item of data that is stored in the spool queue entry when a job is canceled. No action required.

`-NOTIFY has been disabled`

It is not possible to use -NOTIFY and -ON together.

`Notifying semaphore`

Message code 170. Status information only. See the Prime 50 series Technical Summary for information on semaphores No action needed.

`Numeric argument missing`

Message code 74. You have not specified a paper bin number. Specify paper bin number.

`Numeric argument missing after -aaa`

Message code 68. You have omitted the numeric argument from an option in a SPOOL command. *-aaa* is the option that requires the numeric argument. Reissue the command with the numeric argument in place.

`Only one map can be specified using -XLATE`

Message code 86. More than one MAP has been given with the -XLATE option. Check your SPOOL command line.

`-OPEN and -NO_COPY may not be used together`

Message code 89. Reissue the SPOOL command with one or both of the incompatible options removed.

`Opening CONFIG file`

Message code 87. Status message while checking or initializing an environment. No action required.

`Opening EVFU file`
Message code 182. Status message while initializing the EVFU for a printer. No action required.

`Opening Environment File`
Message code 88. Status message while checking or initializing an environment. No action required.

`Opening top-level SPOOL* directory`
Message code 17. Status information only. No action required.

`Opening semaphore`
Message code 91. Information only. A semaphore is a queue of processes where the despooler phantom is queued when it has no work to perform. For details of semaphores see the Prime 50 series Technical Summary. No action required.

`Option -v`
Message code 105. Option has an unexpected qualifier. You have used an argument to an option, where no argument was expected. Issue the command line correctly.

`Option conflict, -DATA_SENSE_ENABLE and -NO_DATA_SENSE_ENABLE`
Message code 142. You have used conflicting options. Reconsider options.

`Option conflict, -DATA_SET_CONTROL and -NO_DATA_SET_CONTROL`
Message code 141. You have used conflicting options. Reconsider options.

`Option conflict, -ECHO and -NO_ECHO`
Message code 143. You have used conflicting options. Reconsider options.

`Option conflict, -ERROR_DETECTION and -NO_ERROR_DETECTION`
Message code 144. You have used conflicting options. Reconsider options.

`Option conflict, -LINE_FEED and -NO_LINE_FEED`
Message code 145. You have used conflicting options. Reconsider options.

`Option conflict, -REVERSE_XOFF and -NO_REVERSE_XOFF`
Message code 146. You have used conflicting options. Reconsider options.

`Option conflict, -XOFF and -NO_XOFF`
Message code 147. You have used conflicting options. Reconsider options.

`Option not recognised`
Message code 121. Mistake in the environment file. Correct the environment file.

`Overlapping margins`

Message code 151. Mistake in the environment file. Correct the environment file.

`Parameter not numeric`

Message code 113. Mistake in the environment file. Correct the environment file

`Pathname not available`

The -SFI option in the SPOOL command prevented the pathname from being displayed. No action required.

`Printer Failure: Data Check.  DPTX`

`Printer Failure: Equipment Check.  DPTX`

`Printer Failure: Mechanically Disabled. DPTX`

`Printer Warning: Carriage Power Off or Print Hung.  DPTX`

`Printer Warning: Printer Timeout.  DPTX`

`Queue file header is corrupt`

Message code 14. The data in the spool queue is corrupt. This could have resulted from a disk error. Follow your installations procedures for recovering from data loss. Restore the Spooler directories to their most recent known state, but retain copies of files that were recently added to the SPOOL_DATA* directories, in case the originals have been deleted from the file system. Notify users of the event, in case they need to reissue some print requests.

`Queue not available at present`

Message code 125. The reason is given in the rest of the message. For example a disk might be shut down. Appropriate action depends on the reason why the queue is not available.

`aaa Reading top-level SPOOL* directory`

Message code 174. A PRIMOS error has occurred while reading the SPOOL* directory. See your system administrator.

`RESET : New environment is aaa`

Confirmation that a PROP command to reset an environment has been successfully obeyed. *aaa* is the name of the new environment. No action required.

`Re-opening log file`

Message code 98. Status information only. Occurs while checking, or initializing, an environment. No action needed.

`Re-setting SPOOL_QUEUE*`

Message code 175. Status information to confirm progress of PROP -COLDSTART. No action required.

`Reading CONFIG file`

Message code 95. Status information only. Occurs while checking, or initializing, an environment. No action required.

`Reading EVFU file`

Message code 183. Status information only. Occurs during initialization of EVFU for a printer. No action required.

`Reading Environment File`

Message code 96. Status information only. Occurs while checking, or initializing, an environment. No action required.

`-RELEASE cannot be applied -NOW`

Message code 172. The -RELEASE option tp PROP only works with a timing of -FINISH or -STOP. Issue the correct PROP command.

`Request nnn added to queue : aaa`

Confirmation that your SPOOL command is successful, and that the print request has been added to a pre-Rev. 21.0 queue. *nnn* is the request number. *aaa* is the queue name. You should note the request number if you think you could need to cancel or modify the request. No action required.

`Request nnn added to queue, nn records : pathname`

Confirmation that your SPOOL command is successful, and that the print request has been added to a Rev. 21.0 queue. *nnn* is the request number. *nn* is the size of the file to be printed (in records). You should note the request number if you think you could need to cancel or modify the request.

`Request nnn added to queue, file unit xx`

Confirms that your SPOOL command, with -OPEN option, is successful and that the print request has been added to a spool queue. Confirms the number of the file unit that has been opened. No action required.

`Request nnn cancelled`

Confirms that your SPOOL command has successfully canceled a print request. No action required.

`Request nnn not found in spool queue`

Message code 99. You have issued a SPOOL command to cancel or modify a print request, but the request could not be found in the queue. This could be because the file has already been printed. If the printed output is not adequate, reissue the print request with the correct attributes and options.

`** Restarted **`

Confirmation that printing has restarted. No action required.

`SET_FONT escape sequence missing`

Message code 56. You have omitted the SET_FONT escape sequence in the environment file. Include the escape sequence.

`-SET_FONT not allowed with pre-rev. 21 queue`

Message code 71. You have used -DISK with -SET_FONT. If you require a pre-rev. 21 queue, you cannot use -SET_FONT.

`-SET_LANDSCAPE not allowed with pr-rev. 21 queue`

Message code 73. You have used -DISK with -SET_LANDSCAPE. If you require a pre-rev. 21 queue, you cannot use -SET_LANDSCAPE.

`-SET_PAPER_BIN not allowed with pre-rev. 21 queue`

Message code 75. You have used -DISK with -SET_PAPER_BIN. If you require a pre-rev. 21 queue, you cannot use -SET_PAPER_BIN.

`-SET_PORTRAIT not allowed with pre-rev. 21 queue`

Message code 76. You have used -DISK with -SET_PORTRAIT. If you require a pre-rev. 21 queue, you cannot use -SET_PORTRAIT.

`Setting RWLOCK on SEM file`

Message code 5. Message from the despooler phantom. Status information only. There is a SEM file for each active despooler phantom to allow communication with the semaphore where the despooler phantom is queued when it has no work to perform. For details of semaphores see the Prime 50 series Technical Summary. No action required.

`-SFI not allowed with pre-rev. 21 queue`

Message code 85. You are trying to use the -DISK and -SFI options together. See your system administrator.

`Sign on message too long`

Message code 131. Mistake in the environment file. Correct the environment file.

`-SPOOL_WHILE_OPEN not allowed with pre-rev. 21 queue`

Message code 78. You are attempting to use -SPOOL_WHILE_OPEN with the -DISK option. Reconsider your options.

`Spooler Failure: Command Reject. DPTX`

Spooler Failure: Operation Check. DPTX

Spooler database has not been initialized

Message code 158. You have tried to use the PROP command to start a despooler phantom, or the SPOOL command to print a file, before initializing the Spooler subsystem. Give the PROP -COLDSTART command.

`Starting up phantom`

Message code 176. Confirmatory message from the PROP command. No action required.

`Stop bits must be 1 or 2`

Message code 139. You have used an invalid qualifier with the -STOP_BITS option of the ASYNC command. Check the environment file.

`** Stopping **`

Confirmation that an environment is stopping as instructed in a PROP command. No action required.

`-SUSPEND must occur -NOW`

Message code 162. The -SUSPEND option to PROP only works with a timing of -NOW. Issue the correct PROP command.

`-TUNIT can only be used with -OPEN`

Message code 106. Your SPOOL command included the -TUNIT option but did not also include the -OPEN option. Reissue the SPOOL command but add the -OPEN option.

`The file used in -NOCOPY must be on the same node as the spool queue`

Message code 82. You have given a SPOOL command with the -NOCOPY option, but the file to be copied is not on the same node as the spool queue to which you directed your request. Either reissue the SPOOL command without the -NOCOPY option, or log in on the system where the file you want to print is located and re-issue the same SPOOL command.

`The filename is too long, 1 to 32 characters please`

Message code 37. When writing a MAP environment statement you have specified too long a string for the filename. Respecify the correct filename.

`The map name is too long, 1 to 32 characters please`

Message code 55. When writing a MAP environment statement you have specified a string that is too long for the map name. Respecify the correct map name.

`The queue is empty`

Status information only, in response to a SPOOL -LIST command. Appropriate action depends on your reason for viewing the spool queue.

`The queue is full`

Message code 94. The SPOOL command is unable to add your request to a pre–Rev. 21.0 queue. Direct the request to a different queue, or wait until some items in the queue are removed.

`This map has been defined on an earlier line.\R\L`

Message code 193. This map name has already been given. Correct the environment file.

`-TO not allowed with pre-rev. 21 queue`

Message code 79. You are attempting to use -TO with the -DISK option. Reconsider options.

`Too Many Command Line Options`

Message code 177. You have given too many options in a PROP command. Issue the correct command. See Appendix A, Summary of Prop and Spool Commands, for details of permitted combinations of options.

`Too many attributes (max 32)`

Message code 150. Mistake in the environment file Correct the environment file.

`Too many device attributes given`

Message code 100. You have given too many attributes in a SPOOL command. Issue a correct command. Details depend on the attributes in use at your installation.

`Too many fonts specified`

Message code 102. You have specified more than one –SET_FONT in a SPOOL command. Retype the SPOOL command.

`Too many MAP commands. The maximum is 64`

Message code 103. You have specified too many MAP commands in an environment file. Reconsider environment file.

`Too many nodes/disks`

Message code 1. Mistake in the environment file. Correct the environment file or use a different environment.

`Too many paper feed devices`

Message code 104. You have stated two sources for your paper when using -SET_PAPER_BIN. Retype -SET_PAPER_BIN.

`Too many paper orientations`

Message code 66. You have specified more than one paper orientation option when using the SPOOL command. Use one only of –SET_LANDSCAPE or –SET_PORTRAIT.

`Trying as rev. 21 style queue (-ON v)`

Status information only, during despooler initialization. No action required.

`Unable to transliterate command line, SIT error code`

Message code 186. The call to SI$TRB has failed. See your system administrator, problem with Standard Internationalization Tools (SIT).

`Unable to transliterate command line, SIT code = $`

Message code 189. The call to SI$TRB has failed. See your system administrator, problem with Standard Internationalization Tools (SIT).

`Unknown device attribute : aaa`

Message code 107. Your SPOOL command contains an attribute that is not allowed on your system, or that is not allowed to your project group or user–ID. *aaa* is the illegal attribute. This message could result from a spelling mistake. Check your SPOOL command and, if necessary, reissue it with the attribute spelled correctly. If the attribute exists, but you are not allowed to use it, see your system administrator.

`Unknown PDN error status nnn`

Message code 126. PDN error is not recognised. Refer to your Network or Device manual.

`Unrecognised Data_Set_Sense`

Message code 135. You have given an incorrect value for the Data_Set_Sense option when using the ASYNC command. Check the environment file.

`Unrecognised parameter type`

Message code 119. Mistake in the environment file. Correct the environment file.

`Unrecognised parity`

Message code 137. You have given an incorrect parity in the ASYNC command. Check the environment file.

`Unrecognised token in EVFU file`

Message code 184. Mistake in the EVFU file. Correct the EVFU file.

`Unsupported character length`

Message code 136. You have given an incorrect qualifier to the -CHAR_LENGTH option of the ASYNC command. Check the environment file.

`Unsupported protocol`

Message code 138. You have given an incorrect qualifier to the -PROTOCOL option of the ASYNC command. Check the environment file.

`Unsupported speed`

Message code 140. You have given an incorrect qualifier to the -SPEED option of the ASYNC command. Check the environment file.

`User data too long`

Message code 133. Mistake in the environment file. Correct the environment file.

`Value out of allowed range`

Message code 108. You have specified a paper bin number outside the range 1-9. Respecify the paper bin number.

`Value out of range in -aaa. Min xx, max yy`

Message code 92. You have given an incorrect numeric argument to an option in a SPOOL command. *-aaa* is the option with the incorrect argument. Reissue the SPOOL command with the numeric argument in the correct range.

`-XLATE not allowed with -PLOT`

Message code 81. You have chosen two incompatible options. Reconsider option.

`-XLATE not allowed with pre-rev. 21 queue`

Message code 80. You are attempting to use -XLATE with the -DISK option. Reconsider the options.

`You do not have clearance to use -SPOOL_WHILE_OPEN`

Message code 51. You are not a user priviledged to use concurrent printing. See your system administrator.

`You do not own this request`

Message code 84. You have attempted to modify another person's spool request. You do not own this request. You have tried to cancel or modify another user's print request. You must be a privileged user to modify other user's requests. The message could result if you typed the wrong request number in the SPOOL command.

Public Data Network Error Messages

```
Access bar
Address is unknown
Bad facility field in users call
Bad parameter(s)
Behind another user in the subprc assign list
Callee refuses collect call
Circuit is cleared
Circuit is idle
Circuit is reset
Clear cause is xxx
DTE busy
DTE clearing
DTE out of order
Destination node not operating
Diagnostic is xxx
Host is down
Illegal address
Illegal or unknown address
Invalid call
Local procedure error
Maximum number of pending operations exceeded
Network congestion
Network server logged out
Networks not configured
No call requests pending
No remote users
No virtual circuits left to accept call
Not enough memory for operation
Not obtainable
Operation complete
Operation illegal now
Operation in progress
Port not assigned
Remote procedure error
Route through protocol error
Route through server down
Routed through call request looping
Routed through circuit timeout
Routed through not enough memory
System not up yet
Timeout on call request
Timeout on clear request
Timeout on reset request
Unknown clearing cause
Unknown diagnostic message
Unknown error status <EEE>
Unknown error status
User does not own virtual circuit
```

D

THE PDN SUPPORT OPTION

Introduction

The PDN Support software allows you to direct your output to printers attached to Public Data Network (PDN) services in the same way as you can direct output to printers attached to asynchronous lines. It is provided as an optional extra feature of the Spooler subsystem.

When Would I Need the PDN Support Option?

Suppose that you have a central office that houses your main computer system, and one or more branch offices that have terminals to the system. The terminals can connect to the computer on dialup connections, or through a PDN such as the American Telenet, the Canadian Datapac, or the British IPSS. The same offices may often need to print files. The Spooler PDN support option allows you to connect a printer at a remote site to a Public Data Network so that it can receive and print data sent from the central system.

To direct a print job to the remote printer you would give a SPOOL command and specify, with the -ATT option, whatever options the system administrator had set up to select the remote printer. This is exactly the same procedure as for any other printer environment, as described in Chapter 1, Introducing the Spooler Subsystem.

If the PDN option were not available you would need to attach the remote printer to an expensive dedicated printer line, or attach a small printer directly to the terminal, and accept a very much slower and poorer quality printing service.

What Other Software is Needed?

To control a printer through a PDN using the Spooler subsystem, you should install PRIMENET with X.25 support. If you have remote terminals that use X.25 multiplexers to attach to the Prime system, or if you have two or more Prime computers in a network, you must already have PRIMENET.

How to Install the PDN Support Option

The supplied tape contains a new library routine and an INFO directory. You should read the files in the INFO directory before installing the new software, in case the version that you obtain needs to be installed differently than described below.

The steps in the installation are as follows:

1. Restore the contents of the tape into a working directory.
2. Copy the file working-dir>LIBRARIES*>SP$PDN_SUPPORT.RUN to LIBRARIES*>SP$PDN_SUPPORT.RUN.
3. Edit the search rules file SPOOL*>ENTRY$.SR to include the line LIBRARIES*>SP$PDN_SUPPORT.RUN. Edit the system search rules file, 'SEARCH_RULES*>ENTRY$.SR, to include LIBRARIES*>SP$PDN_SUPPORT.RUN.

How to Use the PDN Support Option

When the system is correctly set up, the printer environments for remote printers are used in exactly the same way as any other printer environment. The environments are controlled with PROP commands by privileged users of the system. However, if there is no work in any of the spool queues for a despooler phantom that controls a remote printer, the despooler disconnects the PDN link, and becomes idle. When you request a print for the remote printer it is put into a spool queue and the despooler phantom is notified. It then connects to the printer, via the PDN, and remains active until it again has no work to do.

How to Create Printer Environments That use PDN

The despooler phantom that controls a remote printer is controlled with the PROP command in exactly the same way as any other despooler phantom. There is one additional command, PDN, that you must use in the environment file. Apart from this, you create and use the environment file for a remote printer in the same way as any other environment file, as described in Chapter 3, Defining Printer Environments.

Before you can use a printer environment that sends data over the PDN there must be access to the PDN from your system. If there is not, either you, or the system administrator, must use PRIMENET's CONFIG_NET command to configure the PDN. The configuration process determines the synchronous line numbers that correspond to PDN addresses, number of virtual circuits permitted, packet size and so on. For full details see the *Network Planning and Administration Guide.*

If the network is correctly set up you must then find out

- What kind of multiplexer/PAD the remote printer is connected to
- The PDN address of the printer
- Which X.25 facilities you need, or wish to use

The PDN address consists of a maximum of 14 digits and can be given in decimal or octal. It can include a full Data Network Identification Code (DNIC) if necessary. A full PDN address consists of a DNIC field, a network address, and a port-number or sub-address.

In some multiplexers the PDN address is sufficient to connect the host system to the exact port on the multiplexer used by the remote printer. With other multiplexers the PDN address may connect only to the PAD itself, and further information must be supplied to select the port. Sometimes the PAD requires a sign-on-message for security reasons. See NETLINK in the PRIMENET Guide, for further PDN address details.

The PDN Environment File Command

This command is used in environment files for remote printers. It specifies the address of a printer connected to a Packet Assembler/Disassembler (PAD), which in turn is connected to a Public Data Network (PDN). The options are similar to the corresponding options of the NETLINK command. The command is part of the Spooler PDN support option, and is not available with the standard Spooler subsystem.

Command syntax is as follows:

PDN address [options]

You must specify the address in numeric format and it can be a maximum of 14 digits, for example:

```
PDN 63300089 -MPAC 3  (Memotec connected printer on port 3)
```

Options to the PDN command are as follows:

Option	*Description*
-MPAC port	Use this option if the printer is connected to a Memotec 2000 multiplexer. *port* is the one- or two-character Memotec port number.
-GANDALF_PIN service_id **-PIN**	Use this option in the printer is connected to a Gandalf PIN multiplexer. *service-id* is the one- or two-character PIN service ID.
-DYNAPAC port **-DYNA**	Use this option if the printer is connected to a Dynapac X.25 multiplexer. *port* is the Dynapac port number.

The options -MPAC, -PIN, and -DYNA show that a specific type of PAD is being used, and allow the despooler to configure some of the connection parameters automatically.

For any other PDN use the -STANDARD option. Note that if you are connecting to a Telenet PAD, both the PAD and the printer should be configured for no parity. The syntax of the -STANDARD option is as follows:

```
-STANDARD [   -PRID              ]  [ nnn nnn nnn nnn  ]
              -FCTY                 [ nnn ... nnn      ]
              -UDATA                text
            { -SIGN_ON_MSG }        text
            { -SOM         }
```

The way to use the suboptions to the -STD option of the PDN command is outlined below.

Suboption — *Description*

-PRID [*nnn nnn nnn nnn*]

Sets the Protocol Identification field data. It specifies in decimal, the four bytes of the protocol identification field. If the four bytes are omitted, the default, 001 000 000 000, is used. Further details in the *PRIMENET Guide* and your PAD/Multiplexer manual.

-FCTY [*nnn . . . nnn*]

Sets the call request packet facilities. Each parameter is a decimal number in the range 0–255. You can specify up to a maximum of 64 parameters. Parameters are in pairs that correspond to CCITT standards X.3 and X.29. If you give the suboption with no parameters, no facilities are specified. If you omit the -FCTY suboption, the default facilities of 001, 001, 002, and 007 are specified. For more information, and details of facilities codes, see the *PRIMENET Guide.*

-UDATA *text*

Some PADs have several ports, that is, they are multiplexers, but the PDN address in the PDN command only connects to the PAD itself. To select the correct port for the remote printer you must supply some further data. The -UDATA option allows you to supply a string that consists of a maximum of 12 characters. For details of the data to supply with this option see the manual for your Multiplexer or PAD.

-SIGN_ON_MSG *text*
-SOM

This option permits you to supply a sign–on message to the remote PAD/Mutiplexer. Some PADs require the first packet of data received to be a particular sign–on–message. The connection between the host computer and remote printer is made, using the information supplied in the PDN address, and the other options of the PDN commands. The first data packet transmitted is then checked by the PAD/Multiplexer. If the first packet is the correct sign–on–message, the connection continues and further data can be sent to the printer. If the sign–on–message is incorrect the connection is severed. For more information see the manual for your PAD/Multiplexer.

Below is an example of a PDN address that uses some of the -STANDARD suboptions:

```
PDN 123456789012 -STD -PRID 001 000 000 000 -FCTY 12 1 -UDATA 3
```

where:

- -PRID is the standard PRIMENET protocol ID.
- -FCTY turns XON/XOFF.
- -UDATA indicates the port number on the pad you wish to use.

Error Messages

Standard Spooler subsystem messages are listed and explained in Appendix C, Subsystem Messages. When you use the PDN Support option, messages can also arise from the networking software. Such messages are listed in Appendix C, but for full details see the Netlink Guide.

E

DESPOOLING OVER TCP/IP

Introduction

The Transmission Control Protocol/Internet Protocol (TCP/IP) Support software allows you to use the despooler to control printers connected across the Prime TCP/IP network. It is provided as an optional extra feature of the Rev. 22.1 Spooler subsystem.

The printers must be connected to a terminal server or similar device. The address of the terminal server and the port number of the connected printer must be supplied in the despooler environment file.

How to Use the TCP/IP Support Option

When the system is correctly set up, the printer environments for remote printers are used in exactly the same way as any other printer environment. For further details on the use of the Prime TCP/IP software please refer to the PRIMOS TCP/IP Guide.

Refer to the PRIMOS TCP/IP Guide for details on how to configure the port used by the printer. This will be set up as a SLAVE port, typically using XON/XOFF and running at the printer's asynchronous speed (probably 1200 or 9600 baud).

The TCP/IP Environment File Command

This command is used in environment files for remote printers. It specifies the address of the terminal server, and the port number of the connected printer.

Command syntax is as follows:

```
TCP/IP {-ADDRESS a.b.c.d} -PORT port_no [-PAD_LF] {-NAME name_or_alias}
```

Options to the TCP/IP command are as follows:

Option	*Description*
-ADDRESS *a.b.c.d*	*a.b.c.d* is the internet address of the Terminal Server (or another device) connected to the printer. This option cannot be used with the -NAME option. Either -ADD or -NAME must be used.

-PORT *port_no*
Specifies which port the printer is connected to on the Terminal Server. It is not the outgoing port from the host machine.

-NAME *name_or_alias*
name_or_alias is either the name, or a recognized alias, from the local HOSTS file. The despooler attempts to connect to the address associated with the name or alias in the HOSTS file. This option cannot be used with the -ADDRESS option. Either -ADD or -NAME must be used.

-PAD_LF
causes any CR/LF (carriage return/linefeed) pairs to be padded into CR/LF/LF. The TELNET protocol, which most terminal servers adhere to, defines the CR/LF pair be trimmed down to just CR. Hence, use of this option with a Terminal Server that does the trimming ensures that any CR/LF pairs for transmission actually arrive as CR/LF pairs.

It is possible to have more than one despooler use the same printer. Even on the same node it is possible to have two, or more, despoolers attempting to transmit data to the same printer address/port. If two printers come into conflict while trying to connect to an address/port and transmit data, then the second (latest) printer is deferred for five minutes. The mechanism behind the defer is a call to the PRIMOS sleep routine. Any incoming PROP command is not processed while the despooler sleeps. At the end of the sleep the entire reconnection process is invoked again.

INDEX

INDEX

Symbols

.ENV Subscript, 2–6, 2–9, 3–3

.SEM Subscript, 2–7

$REST, 2–3, 2–4

A

Aborting (Despooler Status), 4–2

Access
- SPOOL_DATA* directories, 2–4
- SPOOL_QUEUE* directory, 2–3

Accounting Interface, 2–37
- calling procedure, 2–40
- coding rules, 2–40
- data structure, 2–38

Accounting Records, 2–37

ACLs
- automatic setting, 2–25
- for DESPOOL.CPL, 2–5
- for SPOOL* directory, 2–4
- security, 2–24
- SPOOL_DATA* directories, 2–4
- SPOOL_QUEUE* directory, 2–3

Address Routine, 2–40, 2–41

Aligning Forms, 5–2

ATTRIBUTE Environment Command, 3–4
- -MANDATORY, 3–3, 3–4

Attributes, 1–2
- advanced uses, 2–15
- default values, 2–11, 3–3
- examples in the use of, 2–11
- files, 2–6, 2–11, 2–12, 2–13
- in environment command, 3–4
- mandatory, 2–10
- restricting availability, 2–11
- usage, 2–10
- use of, 3–2
- validation, 2–11

ATTRIBUTES Directory, 2–6, 2–12

attributes files, 2–13

Auxiliary Configuration Parameters, B–9

AUXILIARY Environment Command, B–9

B

Backing Up (Despooler Status), 4–2

Banner Page, 2–41
- interface, 2–42

Banner Page Interface
- coding rules, 2–42
- data structure, 2–42

C

Canceling Print Requests, 4–7—4–9, A–9

Changing a Despooler, 4–5—4–6

Channel Numbers, specifying in EVFU file, 2–31

Character Mapping, 2–5
- ECS_TO_ASCII_MAP, 2–16
- file creation, 2–16
- files, 2–16
- QUME_OAS_MAP, 2–16
- SPOOL*, 2–16

Coding Rules
- accounting interface, 2–40
- banner page interface, 2–42

Cold Start Procedure, 2–29

Commands
  environment file commands, 3–4
  PROP, 2–5
    See also PROP Options
  SPOOL. *See* SPOOL Options

Compatibility, 2–25, 3–6, A–5

CONFIG Environment Command, 3–6

CONFIG QUEUES, 2–6

CONFIG_LIST File, 2–6

Configuration Parameters, AUXILIARY, B–9

Configuring Queue Searching, 2–23, 3–5

Contents of Attributes Files, 2–12

Control Characters
  and EVFU, 2–34
  in printable files, 2–34

Controlling Despooler Phantoms, 4–3—4–5

CPL, for despooler, 2–5

Creation of Environment Files, 3–3

D

Data Structures
  accounting interface, 2–38
  banner page, 2–42

DATA_PARTITIONS, 2–20

DATA_PARTITIONS File, 2–3
  additions and deletions, 2–20

Default Values, of attributes, 3–3

DESPOOL.CPL, 2–5

DESPOOL.RUN, 2–5

DESPOOL_LIBRARY.RUN, 2–27

Despooler Phantoms, 2–7
  at system cold start, 2–30
  changing, 4–5—4–6
  control, 4–3—4–5
  introduction to, 1–4
  names, 3–3, 4–3
  queue searching, 1–4, 3–6
  restarting, 4–4
  starting, 4–3
  status, 4–1
  stopping, 4–5
  suspending, 4–4—4–10
  use of environment files, 3–2

Despooler Status
  aborting, 4–2
  backing up, 4–2
  dropping, 4–2
  hanging, 4–2
  idle, 4–2
  line up, 4–2
  not started, 4–2
  printing, 4–2
  reset, 4–2
  restarting, 4–2
  sleeping, 4–2
  starting up, 4–2
  stopping, 4–2
  SWO, 4–2

DEVICE
  -DICT, 2–29
  -EVFU, 2–31

Device Drivers, AUXILIARY, B–9

Directories, 2–3
  SPOOL*, 2–4
  SPOOL_DATA*, 2–4
  SPOOL_QUEUE*, 2–3

DISK Environment Command, 3–5
  -DECAY, 3–6
  -FAST_INTERVAL, 3–6
  -QUEUE_SEARCH_INTERVAL, 3–6
  -SLOW_INTERVAL, 3–6

DISK_SCAN Environment Command, 3–7

Distribution of Printouts, 2–6, 2–40

Driver Routines Library, 2–5

Dropping (Despooler Status), 4–2

E

E File Prefix, 2–9

EDAC, PRIMOS command, 2–25

Electronic Vertical Format Unit. *See* EVFU

ENTRY$.SR, 2–5

Environment Attributes
  advanced uses, 2–15
  default values, 3–3
  mandatory, 2–10
  usage, 2–10
  use of, 3–2

Environment Commands
  and EVFU files, 2–31
  ATTRIBUTE, 3–3, 3–4
  AUXILIARY, B–9
  CONFIG, 2–6, 3–6
  configuring queue searching, 3–5

DEVICE, 2-29, 2-31
DISK, 3-5
DISK_SCAN, 3-7
FORMAT, 2-31
MAX_SIZE, 3-4
NODE, 2-23, 3-5, 3-7
PDN, D-3
POLL_RATE, 3-8
PRIORITY_SIZE, 3-5
queue search frequency, 3-6, 3-7, 3-8
scheduling print jobs, 3-4
TCP/IP, E-1

Environment Files, 2-6, 2-7, 3-1
alteration, 2-8
command types, 2-9
contents, 4-3
conversion, 2-8
creation, 3-3
example, 3-2
format, 3-4
general restrictions, 3-4
modifying, 4-6
names, 3-3, 4-3
syntax checking, 3-4
usage, 3-2
verification, 4-6
viewing, 2-10

Environment files, modifying, 3-22

Environment Parameters, AUXILIARY, B-9

Environments, 2-7
changing, 4-5—4-6
how many, 2-7
importance of the concept, 1-2
modifying, 4-6—4-7
naming, 3-3
purpose, 1-2
selection, 2-10
special printing requirements, 2-7
status monitoring, 4-1
viewing, 2-10, 4-3

Envrironment Command, MIN_SIZE, 3-5

EPF Library Subroutine, 2-37

EPFs
accounting interface, 2-40
address routine interface, 2-41

EVFU, 2-31
example, action of, 2-34

EVFU Files
alternative formats, 2-31
channel numbers, 2-31
example, new format, 2-32
how used, 2-34—2-43
rules for, 2-31

Examples
action of EVFU, 2-34
new-style EVFU file, 2-32
SPOOL Command, 1-2

F

Files
.ENV, 2-6, 2-9, 3-3
.SEM, 2-7
attributes files, 2-6, 2-11
awaiting printing, 2-20
CONFIG_LIST, 2-6
DATA_PARTITIONS, 2-3
DESPOOL.CPL, 2-5
DESPOOL.RUN, 2-5
DESPOOL_LIBRARY.RUN, 2-5
E, 2-9
ENTRY$.SR, 2-5
environment, 2-7, 3-1
contents, 4-3
creation, 3-3
example, 3-2
format, 3-4
general restrictions, 3-4
names, 3-3, 4-3
syntax checking, 3-4
usage, 3-2
verification, 4-6
viewing, 2-10
where located, 2-6
environment conversion, 2-8
environment file validation, 3-4
environment modifying, 4-6
environment viewing, 2-10
EVFU files, 2-31
in SPOOL* directory, 2-5
in SPOOL_DATA* directories, 2-4
in the SPOOL_QUEUE* directory, 2-3
insert files, 2-6
insert files in SPOOL* > SOURCE, 2-38
maximum size for printing, 3-5
POSTSCRIPT_INIT, 2-5
PRIMOS.COMI, 2-29
printing
overview, 1-1
user action, 1-2
QUEUES, 2-5
search rules, 2-5
SPOOL* > CONFIG_LIST, 2-6
SPOOL* > DESPOOL.CPL, 2-5

  - SPOOL*>DESPOOL.RUN, 2–5
  - SPOOL*>DESPOOL_LIBRARY.RUN, 2–27
  - SPOOL*>ENTRY$.SR, 2–5
  - SPOOL*>POSTSCRIPT_INIT, 2–5
  - SPOOL*>QUEUES, 2–5
  - SPOOL*>USER_ADDRESSES, 2–6
  - SPOOL_QUEUE*>FULL_LIST_USERS, 2–21
  - spooling password protected files, A–1
  - that will not print, 5–1
  - used by the Spooler subsystem, 2–3
  - USER_ADDRESSES, 2–6, 2–40
  - waiting to be printed, 2–4

- FORMAT, 2–31
- Format
  - by EVFU, 2–34
  - environment files, 3–4
- FULL_LIST_USERS File, 2–3, 2–21

H

- Hanging (Despooler Status), 4–2

I

- Identifying Problems, 5–1
- Idle (Despooler Status), 4–2
- Initialization
  - laser printers, 2–28
  - spooler subsystem, 2–30
- Initializtion, EVFU, 2–31
- Installation, tasks involved, 2–2
- Interfaces
  - accounting routine, 2–37
  - banner page routine, 2–42
  - EPF library routine, 2–41

L

- Laser Printers
  - initialization, 2–28
  - PostScript support, 2–28
- Libraries, driver routines, 2–5
- Line Up (Despooler Status), 4–2
- Listing the spool queue, 4–7
- LOG Directory, 2–6

M

- Making a Print Request, A–1
- Mandatrory Attributes, 2–10
- Mapping Files, creation, 2–16
- MAPPINGS, 2–5
  - ECS_TO_ASCII_MAP, 2–5
  - QUME_OAS_MAP, 2–5
  - STANDARD_MAP, 2–5
- MAX_SIZE Environment Command, 3–4
- Mechanical Problems, 5–2
- Messages, C–1
- MIN–SIZE Environment Command, 3–5
- Modifying Print Requests, 4–9, A–9
- Modifying Printer Environments, 4–6
- Monitoring Despoolers, 4–1
- Multi–Language Despoolers
  - -ENV–LANGUAGE, 2–43
  - SIT, 2–43
- Multiple SPOOL_DATA* Directories, 2–20
- Multiplexer, D–3

N

- Naming Conventions
  - attributes files, 2–12
  - environment files, 3–3, 4–3
- Networks
  - access controls, 2–5
  - disks on remote modes, 3–7
  - pre–Rev. 21.0 nodes, 2–26
  - Public Data Network (PDN), D–3
  - rings, 3–7
- NODE
  - -LOCAL, 3–5
  - -RING, 2–23, 3–5
- NODE Environment Command, 3–5, 3–7
  - -DECAY, 3–8, 3–9
  - -FAST_INTERVAL, 3–8
  - -LOCAL, 3–7
  - -PRE_REV_21, 3–7
  - -QUEUE_SEARCH_INTERVAL, 3–8
  - -RING, 3–7
  - -SLOW_INTERVAL, 3–8
- Nodes

remote disks, 3-7
remote spool queues, 3-6

Not Started (Despooler Status), 4-2

O

Operator, procedures, 4-1

Operators, routine tasks, 1-6

Output Devices
laser printers, 2-28
print handlers, 2-27

P

Packet Assemblers/Disassemblers (PAD), D-3

Password Protected Files, A-1

PDN Environment Command, D-3
-DYNAPAC, D-3
-GANDALF_PIN, D-3
-MPAC, D-3
-STANDARD, D-3

Phamtoms. *See* Despooler Phantoms

phantoms. *See* Despooler Phantoms

POLL_RATE Environment Command, 3-8

Polling Spool Queues
environment commands, 3-5
specifying frequency, 3-6, 3-7

PostScript, 2-28
and PROP -BACK, 5-3

POSTSCRIPT_INIT File, 2-5

Preparation for Installation, 2-2

PRIMENET, rings, 3-7

PRIMOS Commands
EDAC, 2-25
SAC, 2-25

Print Handler, 2-27

Print Handlers, B-1
AUXILIARY, B-9

Print Requests
canceling, A-9
checking progress, 1-3
correcting mistakes, 1-3
examples, 1-2
modifying, 4-9
on remote systems, 2-26

Printer, jams, 5-2

Printer environment files, modifying, 3-22

Printer Environments, 2-7—2-43
importance of the concept, 1-2
modifying, 4-6—4-7
number, 2-7
purpose, 1-2
selection, 2-10
special printing requirements, 2-7

Printers
driver routines, 2-5
environment, 3-1
handler routines, B-1
initialization
EVFU, 2-31
laser, 2-28
print handlers, 2-27

Printing
Banner Page, 2-41
canceling, 4-7—4-9
canceling print requests, A-9
file size, 3-5
hastening, request, 4-9
making print requests, 1-1, A-1
modifying print requests, A-9
overview of process, 1-1
prerequisites, 1-5
software process, 1-4
specifying requirements, 3-2
user actions, 1-2

Printing (Despooler Status), 4-2

Printout Distribution, 2-40

PRIORITY_SIZE Environment Command, 3-5

Privileged Users
facilities available, 2-25
modifying print requests, A-9
viewing the spool queue, A-10

Problems
file will not print, 5-1
mechanical, 5-2
printer hangs, 5-2
printer jam, 5-2
unfamiliar problems, 5-1

Procedures
aligning forms, 5-2
changing a despooler, 4-5—4-6
operator routines, 4-1
restarting a despooler, 4-4
starting a despooler, 4-3—4-10
stopping a despooler, 4-5
suspending a despooler, 4-4
system cold start, 2-29

PROP, -ENV_LANGUAGE, 2-43

PROP Options
  -ABORT, 4-8, 5-3, A-13
  -BACK, 5-3, A-14
  -COLDSTART, 2-30, A-14
  -CONTINUE, 4-4, 5-2, A-14
  -DISPLAY, 2-10, 4-3, 5-2, A-14
  -DROP, 4-8, A-14
  -ENV_LANGUAGE, A-14
  -HANG, 4-4, A-14
  -HELP, A-14
  -LINEUP, 5-2, A-14
  -RELEASE, 4-8, A-14
  -RESET, 4-6, A-15
  -RESTART, 5-3, A-15
  -START, 2-30, 4-3, 4-6, 5-2, A-15
  -STATUS, 4-1, 5-2, A-15
  -STOP, 4-5, A-15
  -SUSPEND, 4-8, A-15
  -VERIFY, 3-4, 3-5, A-15
  -verify, 4-6
  summary, A-12

PROP Syntax, A-12

Public Data Network, D-3

Q

QUEUES File, 2-5

R

Remote Disks, reconstruction of lists, 3-6

Reset (Despooler Status), 4-2

Restarting (Despooler Status), 4-2

Restarting a Despooler, 4-4

RESTRICTED_USERS, 2-6
  SWO_VALID_USERS, 2-6

Restrictions on Environment Files, 3-4

Ring Networks, 3-7

Routine Operational Tasks, 1-6

Runfiles, for SPOOL and PROP, 2-3

S

SAC, PRIMOS command, 2-25

Scheduling Print Jobs, environment commands, 3-4

Search Rules
  ENTRY$.SR, 2-5
  for accounting routine, 2-40
  for address routine, 2-41
  for banner page routine, 2-42
  for display spool queue routine, 2-22

Searching Spool Queues, 3-5
  frequency, 3-6
  specifying frequency, 3-7

Security
  -NO_COPY options of SPOOL, 2-24
  ACLs, 2-24
  C2 classification, 2-24
  privileged users, 2-25
  SPOOL$$ group, 2-24
  SPOOL_DATA* directories, 2-24
  spooling password protected files, A-1

Setting up the Spooler Subsystem, details, 2-1

Setting up the Spooler System, overview, 1-5

Sharing the Spooler Subsystem, 2-29

Sleeping (Despooler Status), 4-2

SOURCE Directory, 2-6

SPOOL and PROP runfiles, 2-3

SPOOL Command
  -NO_FORMAT, 2-34
  canceling requests, A-9
  checking job progress, 1-3
  coreecting mistakes, 1-3
  examples, 1-2
  general description, 1-1, A-1
  making a print request, A-1
  modifying print requests, 4-9
  software process, 1-4
  syntax, A-1
  user's view, 1-1
  viewing the spool queue, A-10

SPOOL Options
  -ALIAS, 2-41, A-4
  -AS, A-4
  -AT, 2-11
  -ATTRIBUTE, 2-11, A-5
  -CANCEL, 4-8, A-5, A-9
  -COBOL, A-5
  -COPIES, A-5
  -DEFER, A-5
  -DISK, 2-19, 2-26, A-5
  -FORM, 2-11
  -FROM -TO, A-6
  -FTN, A-6
  -HEADER, A-6
  -HELP, A-6
  -LIST, 1-3, 4-7, A-6

-LNUMBERS, A-6
-MODIFY, 4-9, A-6
-NO_COPY, 1-4, 2-4, 2-20, A-7
security aspects, 2-24
-NO_EJECT, A-7
-NO_FORMAT, A-7
-NO_HEADER, A-7
-NO_SWO, A-7
-NOP, A-7
-NOTIFY, A-7
-NPH, A-7
-ON, 2-19, A-7
-OPEN, A-7
-PLOT, A-8
-PROC, A-8
-RUSH, 4-9, A-8
-SET_FONT, A-8
-SET_LANDSCAPE, A-8
-SET_PAPER_BIN, A-8
-SET_PORTRAIT, A-8
-SFI, A-8
-SPOOL_WHILE_OPEN, A-8
-SWO, 2-4, 2-20
-TRUNCATE, A-9
-TUNIT, A-9
_TYPE, 2-11
summary, A-2

Spool Options, -PROC, 2-28

Spool Queues, 1-3
adding a request, A-1
canceling requests, A-9
configuring queue searching, 3-5
defining polling rates, 2-24
management, 2-5
modifying a request, A-9
on other nodes, 2-23, 2-26, 3-7
pre-Rev. 21, 3-6
search configuration, 2-23
search frequency, 2-24, 3-7
setting up, 2-19
specifying frequency, 3-6
viewing, 2-3, 2-21, 4-7, A-10

Spool queues, viewing, 2-5

SPOOL Syntax, A-1
canceling print requests, A-9
modifying a print request, A-9

SPOOL$$, 2-3, 2-4
security, 2-24

SPOOL* Directory, 2-4

SPOOL* > ATTRIBUTES Directory, 2-6, 2-12

SPOOL* > DESPOOL_LIBRARY.RUN, 2-5

SPOOL* > ENTRY$.SR, alter search rules for new EPF, 2-37

SPOOL* > LOG Directory, 2-6, 2-10

SPOOL* > SOURCE Directory, 2-6

SPOOL* > USER_ADDRESSES File, 2-6

SPOOL_ADMINISTRATOR$, facilities available, 2-25

SPOOL_DATA* Directories, 2-4
on more than one partition, 2-20
security, 2-24

SPOOL_QUEUE* Directory, 1-4, 2-3

SPOOL_QUEUE* directory, security, 2-24

Spooling a File, A-1
checking the spool queue, 1-3, A-10
examples, 1-2
modifying a request, A-9
software process, 1-4

Starting a Despooler, 4-3—4-10

Starting Up (Despooler Status), 4-2

Status, environment/despooler, 4-1

Stopping (Despooler Status), 4-2

Stopping a Despooler, 4-5

Sub-directories
SPOOL* > ATTRIBUTES, 2-6
SPOOL* > LOG, 2-6, 2-10
SPOOL* > SOURCE, 2-6

Summaries
PROP Options, A-12
SPOOL Options, A-2

Suspending a Despooler, 4-4

SWO (Despooler Status), 4-2

Syntax
environment files, 3-4
PROP Command, A-12
SPOOL Command, A-1, A-9, A-10

System Administrator, 2-1
access to SPOOL* directory, 2-4
install preparations, 1-5

System Cold Start, 2-29

System Search Rules, 2-5

T

TCP/IP Environment Command, E-1
-ADDRESS, E-1

-NAME, E-2
-PAD_LF, E-2
-PORT, E-2

Transmission Control/Internet Protocol (TCP/IP), E-1

U

USER_ADDRESSES File, 2-40

Users
actions to print file, 1-1
address for printout distribution, 2-40
SPOOL Command, user's view, 1-1
Spooler Subsystem, user's view, 1-1

Utilities
ACL installation, 2-25
environment file conversion, 2-8

V

Validation
attributes, 2-11
environment files, 3-4

Viewing
environments, 4-3
spool queues, 2-3, 2-21, A-10

Summaries
PROP Options, A-13
SPOOL Options, A-2

Suspending a Despooler, 4-4

SWO (Despooler Status), 4-2

Syntax
environment files, 3-4
PROP Command, A-13
SPOOL Command, A-1, A-10, A-11

System Administrator, 2-1
access to SPOOL* directory, 2-4
install preparations, 1-5

System Cold Start, 2-29

System Search Rules, 2-5