

ULTRIX

Guide to PrintServer Clients

Order Number: AA-PC31B-TE

December 1991

Product Version:

ULTRIX Version 4.2A

This manual explains how to use the PrintServer Client for ULTRIX software on ULTRIX operating systems to print files on Digital's family of PrintServer network laser printers.

digital equipment corporation
Maynard, Massachusetts

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

© Digital Equipment Corporation 1988, 1989, 1990, 1991
All rights reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation 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.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

ALL-IN-1, Bookreader, CDA, DDIF, DDIS, DEC, DECnet, DECstation, DECsystem, DECUS, DECwindows, DTIF, MASSBUS, MicroVAX, Q-bus, ULTRIX, ULTRIX Mail Connection, ULTRIX Worksystem Software, UNIBUS, VAX, VAXstation, VMS, VT, XUI, and the DIGITAL logo.

PostScript and Display PostScript are registered trademarks of Adobe Systems, Inc. Tektronix is a trademark of Tektronix, Inc. UNIX is a registered trademark of UNIX System Laboratories, Inc.

Contents

Preface	vii
---------------	-----

1 The PrintServer Printing System

1.1	Client	1-2
1.1.1	Line Printer Daemon	1-3
1.1.2	Device Control Library	1-4
1.1.3	Translators	1-4
1.1.4	Communications Filters	1-4
1.2	Supporting Host	1-5
1.3	PrintServer Family	1-6
1.3.1	The PrintServer 20 Hardware	1-7
1.3.1.1	Print Engine	1-8
1.3.1.2	Controller	1-9
1.3.2	The PrintServer 40 Series Hardware	1-10
1.3.2.1	Print Engine	1-11
1.3.2.2	Data Controller	1-12
1.3.2.3	Device Controller	1-13
1.3.2.4	Console	1-13
1.3.3	Printing Services Software	1-13
1.4	Local and Remote Printing	1-14
1.5	PrintServer Configurations	1-15
1.5.1	TCP/IP Network	1-15
1.5.2	DECnet Network	1-16

2 Printing on a PrintServer System

2.1	Overview	2-1
2.2	lpr Command Options	2-2
2.2.1	Syntax	2-2
2.2.2	Defaulting Scheme	2-3
2.2.3	data_type	2-3
2.2.4	input_tray	2-4
2.2.5	layup_definition	2-5
2.2.6	message	2-6
2.2.7	number_up	2-6
2.2.7.1	number_up Option and Page Orientation	2-8
2.2.7.2	number_up Option Paper Selection Options	2-8
2.2.8	orientation	2-9
2.2.8.1	orientation with number_up	2-10
2.2.9	output_tray	2-10
2.2.10	page_limit	2-13
2.2.11	page_size	2-13
2.2.12	sheet_count	2-14

2.2.13	sheet_size	2-14
2.2.14	sides	2-15
2.2.14.1	sides Option with Page Orientation	2-21
2.2.15	Error Messages	2-21
2.3	ULTRIX Print Queue Commands	2-23
2.3.1	lpq	2-23
2.3.2	lprm	2-24
2.3.3	lpc	2-24
2.4	Setting Up POSTSCRIPT Print Jobs	2-24
2.4.1	Creating Setup Modules	2-25
2.4.2	Sample Setup Modules	2-25
2.4.3	Translating ANSI Files into POSTSCRIPT	2-26
2.4.3.1	Examples	2-26
2.4.4	ARP Entry for Printing Over a TCP/IP Network	2-26
2.4.5	Unknown Message from TCP/IP PrintServer	2-27
2.4.6	ANSI Preamble Loading for TCP/IP PrintServers	2-27

3 Using Layup Definition Files

3.1	Creating a Layup Definition File	3-1
3.1.1	Borders	3-2
3.1.2	Sheet Margins	3-2
3.1.3	Margin Alternation	3-6
3.1.4	Pages per Sheet	3-6
3.1.5	First Page	3-6
3.1.6	Page Order	3-7
3.1.7	Page Grid	3-8
3.2	Sample Layup Definition Files	3-8

4 Device Control Library

4.1	Device Control Modules	4-1
4.2	Order of Device Control Modules in a Print Job	4-3
4.3	Creating a Device Control Module	4-3

5 Media Handling

5.1	Letterhead Paper	5-1
5.1.1	PrintServer 20 Network Printer	5-1
5.1.2	PrintServer 40 Series Network Printers	5-1
5.2	Drilled Paper	5-1
5.2.1	PrintServer 20 Network Printer	5-1
5.2.2	PrintServer 40 Series Network Printers	5-2
5.3	Transparencies	5-2
5.3.1	PrintServer 20 and PrintServer 40 Series Network Printers	5-2
5.4	Gummed Labels	5-2
5.4.1	PrintServer 20 and PrintServer 40 Series Network Printers	5-2
5.5	Preprinted Forms	5-2

6 Error Logging and Accounting

6.1	Error Logging	6-1
6.1.1	Message Format	6-1
6.1.2	Message Descriptions	6-1
6.1.2.1	BADLAYDEF Messages	6-10
6.2	Accounting	6-14
6.2.1	Job Flag Page	6-15

Glossary

Index

Examples

2-1	Sample POSTSCRIPT Code to Change from Portrait to Landscape Orientation	2-10
-----	--	------

Figures

1-1	PrintServer Printers in a Network	1-1
1-2	PrintServer DECnet Printing System	1-2
1-3	PrintServer TCP/IP Printing System	1-2
1-4	PrintServer Client Software on an ULTRIX System	1-5
1-5	PrintServer 20 Hardware Components	1-8
1-6	PrintServer 20 Data Controller	1-10
1-7	PrintServer 40 Series Hardware Components	1-11
1-8	PrintServer 40 Data Controller	1-12
1-9	Local and Remote Printing on a DECnet Network	1-15
1-10	Local and Remote Printing on a TCP/IP Network	1-15
1-11	Sample TCP/IP PrintServer Configuration	1-16
1-12	Sample DECnet PrintServer Configuration	1-17
2-1	Four Pages to a Sheet (4-Up)	2-7
2-2	Page Orientation with number_up	2-8
2-3	Landscape Pages on a Portrait Sheet	2-11
2-4	Sample Layout for One-Sided Printing	2-16
2-5	Sample Layout for Two-Sided Printing	2-17
2-6	Sample of Two-Sided Printing (Portrait Page)	2-18
2-7	Sample of Two-Sided Printing (Landscape Page)	2-18
2-8	Sample of Tumble Printing (Portrait Page)	2-19
2-9	Sample of Tumble Printing (Landscape Page)	2-20
3-1	Four Pages to a Sheet (4-Up) with Borders	3-3
3-2	Four Pages to a Sheet (4-Up) Without Borders	3-4
3-3	Usable Sheet Area with Margins Option	3-5
3-4	Examples of Page Order Option (-N9)	3-7
6-1	Sample Job Flag Page	6-16

Tables

1-1	Comparison of PrintServer Printing Features	1-7
2-1	lpr Options for PrintServer Printers	2-3
2-2	data_type Qualifiers	2-4
2-3	PrintServer Input Trays	2-4
2-4	message Qualifiers	2-6
2-5	orientation Qualifiers	2-9
2-6	output_tray Qualifiers	2-12
2-7	Valid output_tray Qualifiers on PrintServer Models	2-12
2-8	page_size Qualifiers	2-14
2-9	sheet_size Qualifiers	2-14
2-10	sides Option Qualifiers	2-20
3-1	Layup Definition File Options	3-2
3-2	Values for the PageOrder Option	3-7
4-1	Order of Data Sent to a Print Job	4-3

Preface

This manual explains how to use the PrintServer Client for ULTRIX software on ULTRIX operating systems to print files on Digital's family of PrintServer network laser printers.

Intended Audience

This manual is written for users and system administrators of the PrintServer Client for ULTRIX system. You should be familiar with the ULTRIX operating system.

Document Structure

The manual contains the following chapters:

- Chapter 1 provides an overview of Digital's PrintServer network printers.
- Chapter 2 explains the commands needed to print files and manage queues on a PrintServer Client for ULTRIX system.
- Chapter 3 explains how to create and use layout definition files to control the mapping of pages to sheets.
- Chapter 4 describes the functions of the POSTSCRIPT device control library modules and explains how to use these modules to customize a print job.
- Chapter 5 discusses how to handle some special types of media on a PrintServer printer system.
- Chapter 6 describes error logging and accounting on a PrintServer system.
- The Glossary defines technical terms used in this manual.

Associated Documents

The following documents are also associated with Digital's PrintServer systems:

VAX PrintServer Manuals

- *Installation Guide: VAX PrintServer Supporting Host*
- *Management Guide: VAX PrintServer Supporting Host*
- *Installation Guide: VAX PrintServer Client*
- *Management / User's Guide: VAX PrintServer Client*

POSTSCRIPT Printer Manuals

- *Programmer's Supplement: POSTSCRIPT Printers*
- *POSTSCRIPT Translators Reference Manual*

POSTSCRIPT Books by Adobe Systems, Incorporated

- *POSTSCRIPT Language Reference Manual*
- *POSTSCRIPT Language Tutorial and Cookbook*
- *POSTSCRIPT Language Program Design*¹

ULTRIX Manuals

- *ULTRIX Reference Pages*
- *ULTRIX Supplementary Documents, Volume 2: Programmer – 4.2BSD Line Printer Spooler Manual*

Conventions

The following conventions are used throughout this manual:

Convention	Meaning
%	The default user prompt is your system name followed by a right angle bracket. In this manual, a percent sign (%) is used to represent this prompt.
#	A number sign is the default superuser prompt.
UPPERCASE	The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown.
<i>filename</i>	In examples, syntax descriptions, and function definitions, italics are used to indicate variable values; and in text, to give references to other documents and to introduce new terms that are defined in the Glossary.
[]	In syntax descriptions and function definitions, brackets indicate items that are optional.
...	In syntax descriptions and function definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
.	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.
cat(1)	Cross-references to the <i>ULTRIX Reference Pages</i> include the appropriate section number in parentheses. For example, a reference to cat(1) indicates that you can find the material on the cat command in Section 1 of the reference pages.
Boldface	ULTRIX commands, file names, directory names, script names, and program names are boldfaced in text.
!	Comment lines in examples begin with an exclamation point.
<i>italics</i>	Italics are used to introduce new terms, which are defined in the Glossary, and variable items in syntax.

Note

Every effort has been made to replicate the appearance of terminal displays in the printed examples contained in this manual. In some cases, however, lines must divide in places where they do not break in the on-screen display in order to accommodate the margins of this book.

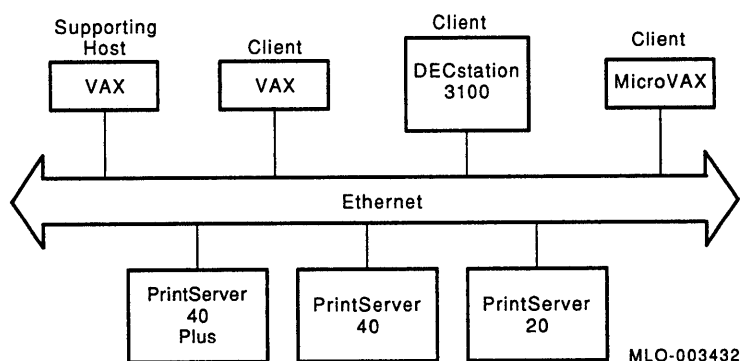
¹ This book cannot be ordered through Digital's United States Software Supply Business.

The PrintServer Printing System

This chapter describes the PrintServer printing system, which consists of three main components: client, supporting host, and printer.

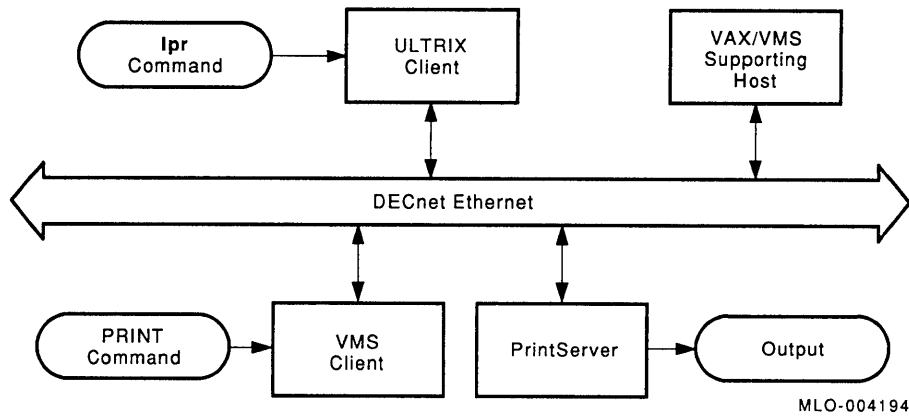
Client software relays information between a client and a *PrintServer printer* over a DECnet or a TCP/IP *network*. Client software requires supporting host software to start and manage the printer. Each client, supporting host, and PrintServer printer is a *node* on the network. See Figure 1-1.

Figure 1-1 PrintServer Printers in a Network



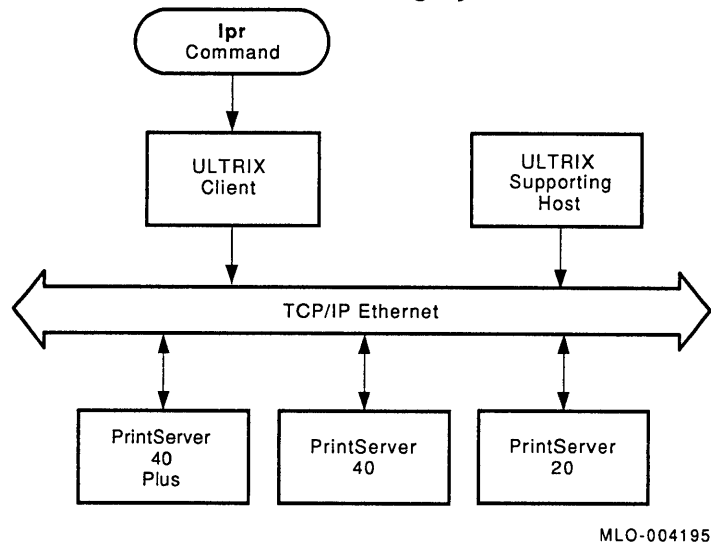
The client, supporting host, and the printer form a *printing system*. Client software allows you to print to PrintServer printers in two different network configurations: the *DECnet* network and the *TCP/IP* network. Client software on a DECnet network requires a VMS supporting host to boot and manage the PrintServer network printer. Figure 1-2 shows a PrintServer DECnet printing system.

Figure 1–2 PrintServer DECnet Printing System



Client software on a TCP/IP network requires a VAX ULTRIX supporting host to boot and manage the PrintServer network printer. Figure 1–3 shows a PrintServer TCP/IP printing system.

Figure 1–3 PrintServer TCP/IP Printing System



The rest of this chapter provides details about the client, the supporting host, and the PrintServer family.

1.1 Client

A PrintServer *client* is a node running either the ULTRIX or VMS operating system, from which users can send print requests to a PrintServer. Client software installed on the node handles users' print requests and provides direct communication with the printer. A PrintServer system can have any number of clients. The client software is available to all users on a client system.

Refer to the *Management/User's Guide: VAX PrintServer Client* for a description of the VMS client.

The PrintServer Client for ULTRIX software consists of the following components, which are discussed more fully in this chapter:

- A user interface with **lpr** command options that allows you to use the features of a PrintServer system and queue management commands. (Chapter 2 explains how to use these commands with a PrintServer system.)
- A line printer daemon (**lpd**) that processes print requests from the ULTRIX **lpr** command.
- A device control library, containing modules of POSTSCRIPT procedures. The library modules allow the POSTSCRIPT interpreter to provide features such as options to the **lpr** command. (See the *Programmer's Supplement: POSTSCRIPT Printers* for more information on writing POSTSCRIPT programs for Digital printers.)
- Translators that convert a data format into the POSTSCRIPT page description language. These translators include ANSI, ASCII, ReGIS, and Tektronix 4010/4014. Another translator converts layout definition files into POSTSCRIPT. (See the *POSTSCRIPT Translators Reference Manual* for more information on POSTSCRIPT translators.)
- Filters executed by the line printer daemon, which passes them POSTSCRIPT data. Filters contain distribution software that transmits print requests over the DECnet or TCP/IP network to the printer and relays system messages from the printer to the line printer daemon.

1.1.1 Line Printer Daemon

The *line printer daemon* (**lpd**) is the spool area handler. The **lpd** command receives requests to print, display, and remove files in the print queue. It also starts print filters for specific print requests.

The **lpd** process is generally started at boot time from the **/etc/rc.local** file. The **lpd** command invokes the line printer daemon. When users submit print jobs with the **lpr** command, the line printer daemon schedules jobs to be printed and notifies printers that have jobs waiting.

When first invoked, the line printer daemon checks the *spooling directory* for each PrintServer device for the existence of a *lock file*. The lock file is a signal to **lpd** that another job is currently printing. If a lock file is not present, **lpd** creates one to reserve access to the printer for that particular print job.

After the daemon has successfully created the lock file, it scans the directory for command files with names beginning with **cf**. These files specify names of files that are to be printed and parameters affecting how the files are to be printed. The data file beginning with **df** contains the untranslated text to be printed.

The line printer daemon uses the command file along with default information from a printer description file called **/etc/printcap** to print the job when the printer is free.

The line printer daemon can mail error messages to the user, if specified either in the **lpr** command line or the **/etc/printcap** file. The line printer daemon also stores printer accounting and operator error log information in files, if specified in the **/etc/printcap** file.

For further information, see **lpd(1)** in the *ULTRIX Reference Pages*

1.1.2 Device Control Library

A PrintServer client's *device control library* is an ULTRIX archive file that contains modules written in POSTSCRIPT. Each client has a device control library.

The line printer daemon uses modules from the device control library to implement various **lpr** command line options that affect output appearance. The line printer daemon extracts the appropriate module, depending on the option that appears on the **lpr** command line, and inserts the module into the data stream of the print job.

The device control library also contains modules that **lpd** does not automatically use. Some of these modules contain POSTSCRIPT procedures, for example, to include an error handler in a print job or to extend the **findfont** procedure to recognize alternative character encodings.

1.1.3 Translators

Because each job a client sends to the printer must be in POSTSCRIPT format, **lpd** can send a job's print data to a translator to change the file's data format into POSTSCRIPT. A *translator* is a stored program that translates or converts a particular imaging data syntax, such as ANSI or ReGIS, into the POSTSCRIPT language. For example, suppose you specify the following **lpr** command:

```
% lpr -Pscript -Dans i filename
```

lpd reads the information from the **lpr** command options and sends the ANSI input data in *filename* to the ANSI translator to change the ANSI input data to POSTSCRIPT.

If you need details about translators, see the *POSTSCRIPT Translators Reference Manual*.

1.1.4 Communications Filters

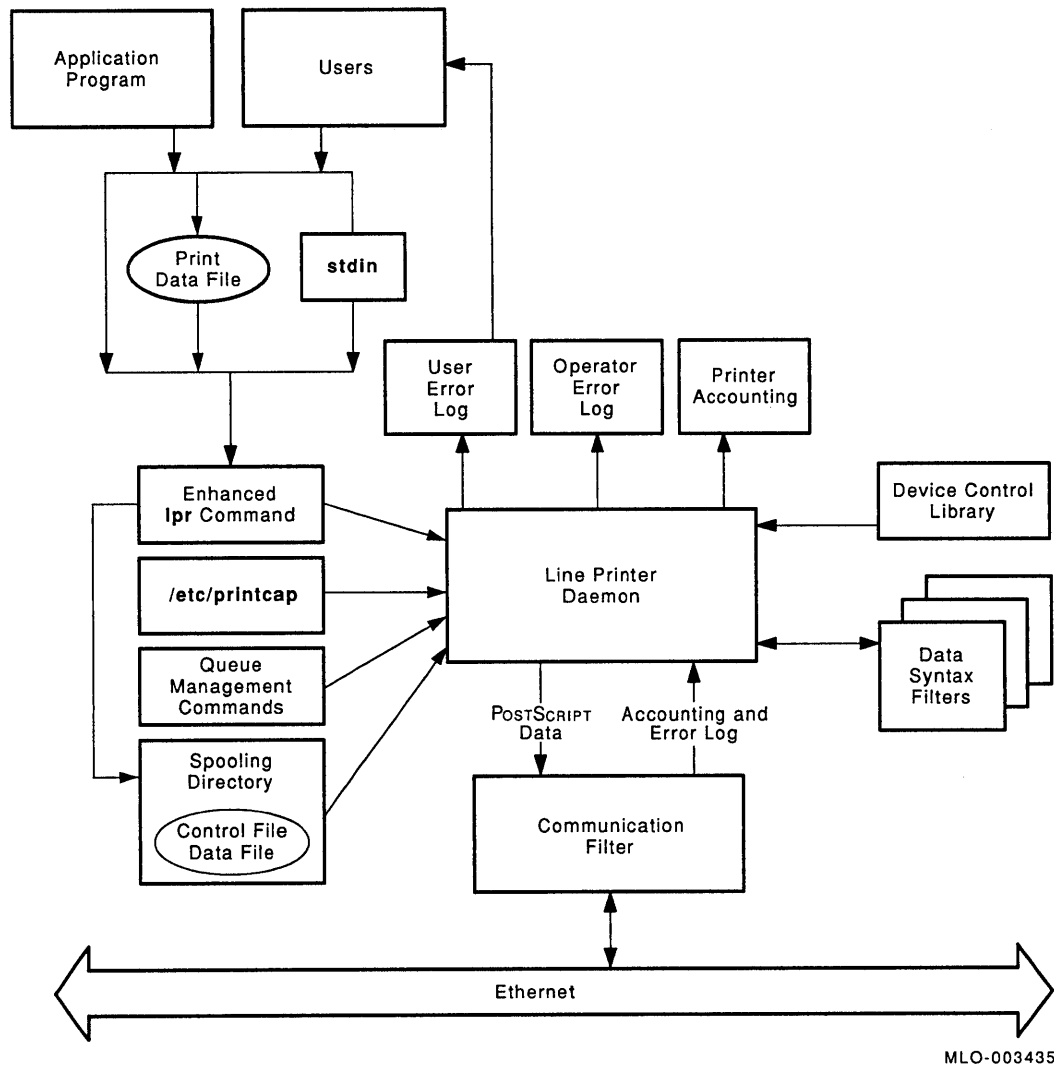
The line printer daemon executes a communications *filter* to which it passes POSTSCRIPT data of a print job. Communications filters transmit print requests over the DECnet or TCP/IP network to a PrintServer device and relay accounting information as well as user and operator error logs. Client software includes two communications filters for transferring data across the network:

- The **lpscomm** filter contains the DECnet distribution software.
- The **iplpscomm** filter contains the TCP/IP distribution software.

Communications filters also return accounting and error log information to their standard error. The line printer daemon executes a filter called **lpserroff**, which stores operator error logs and printer accounting information in separate files, if specified in the **/etc/printcap** file. With the message **-M** option specified, the line printer daemon mails the error log information to the user.

Figure 1-4 illustrates the process by which print files are produced on the ULTRIX PrintServer Client system and delivered to the printer.

Figure 1-4 PrintServer Client Software on an ULTRIX System



1.2 Supporting Host

The *supporting host* is a VAX/VMS system for a DECnet network or a VAX/ULTRIX system for a TCP/IP network designated to start up and manage the PrintServer system. The *Printing Services Software* resides on the supporting host and is *down-line loaded* when the printer is booted. (See Section 1.3.3 for a description of the Printing Services Software.)

Supporting host software creates and maintains a *configuration database* and event log for the PrintServer device and provides *server management services* through an interface, which you can use from terminals on any supporting host in the network. The only requirement is that the PrintServer node be known to the supporting host's network database. You can also manage the servers at the PrintServer 40 or PrintServer 40 Plus console.

Only one supporting host is necessary in a PrintServer system. Server managers can set up other machines as backup supporting hosts, however, so that if one is not available to load the image file during printer boot, another host can perform this function.

When a PrintServer printer is powered up, it broadcasts a load request on the *Ethernet network*. The first supporting host system that answers the request and down-line loads the printer becomes the *primary supporting host* for that printer. The PrintServer printer accesses its configuration database and stores its *event-logging* data on the primary supporting host.

VMS Supporting Host in a DECnet Network

A VAX supporting host can support any number of PrintServer printers. A DECnet PrintServer printer can have only one VMS primary supporting host at a time. A system manager can set up other systems as backup supporting hosts to ensure that the image file loads when the printer boots. A system manager can also change the primary VMS supporting host. Refer to the *Management Guide: VAX PrintServer Supporting Host* for a description of the VMS host.

ULTRIX Supporting Host in a TCP/IP Network

An ULTRIX supporting host can support any number of PrintServer printers. A TCP/IP PrintServer printer can have any number of ULTRIX supporting hosts. One supporting host can handle all administrative functions for one or more PrintServer printers. Or you can distribute these functions among different supporting hosts. Refer to the *User/Management Guide: PrintServer TCP/IP Supporting Host for ULTRIX* for a description of the ULTRIX host.

1.3 PrintServer Family

The PrintServer family is a group of compatible, high-quality laser printers that allow you to share printing resources on a network. This family includes the PrintServer 20 model, which prints at speeds of up to 20 pages per minute, and the PrintServer 40 models, which print at speeds of up to 40 pages per minute.

There is a basic PrintServer 40 model, which contains a Micro/VAX II processor and a PrintServer 40 Plus, an enhanced model that contains a CVAX processor. PrintServer 40 series printers refers to the PrintServer 40 and the PrintServer 40 Plus network printers.

The PrintServer family of printers are high-resolution, high-speed printers that use a common page description language, POSTSCRIPT. They can print pages that integrate both text and graphics and they can print a variety of text *fonts*.

The PrintServer family is intended for business, scientific, and technical customers, in-house publishers, and electronic data processing departments for high-quality printing of mixed text and graphics.

Each PrintServer printer addresses the need for different printing speeds and volumes. The PrintServer 20 model is well suited to the office cluster printing environment. The PrintServer 40 model may better serve larger satellite networks, such as entire departments.

Each PrintServer contains software to receive incoming POSTSCRIPT-formatted print files, a controller to process POSTSCRIPT file data (page descriptions) into *bitmap* images, and a print engine to produce the marks on cut sheets of paper and transparencies.

The following print features are common to all members of the PrintServer family:

- ANSI/Sixel, ASCII, Tektronix 4010/4014, and ReGIS data syntaxes
- DECnet network/TCP/IP network
- Paper sizes: Letter or A, A3, A4, A5; Ledger or B, B4, B5; Executive; and Legal
- POSTSCRIPT interpreter
- 29 resident fonts
- 300 dots-per-inch resolution

Table 1–1 compares the printing features of the PrintServer 20 and PrintServer 40 models.

Table 1–1 Comparison of PrintServer Printing Features

Feature	PrintServer 20	PrintServer 40
Duplex mode	Yes	No
Input trays	Two 250-sheet capacity; one 2000-sheet LCIT ¹	Two 250-sheet capacity upper and lower trays; one 2000-sheet LCIT
Offset stacking	Yes	No
Operator console	Panel on printer	Separate VT320
Output trays	Two upper and lower 750-sheet job separation, face-down stackers; one 100-sheet side tray	Two 500-sheet capacity stackers (only top stacker is face down)
Print speed (simplex mode ²)	Up to 20 images/min	Up to 40 images/min

¹Large capacity input tray
²One-sided printing

1.3.1 The PrintServer 20 Hardware

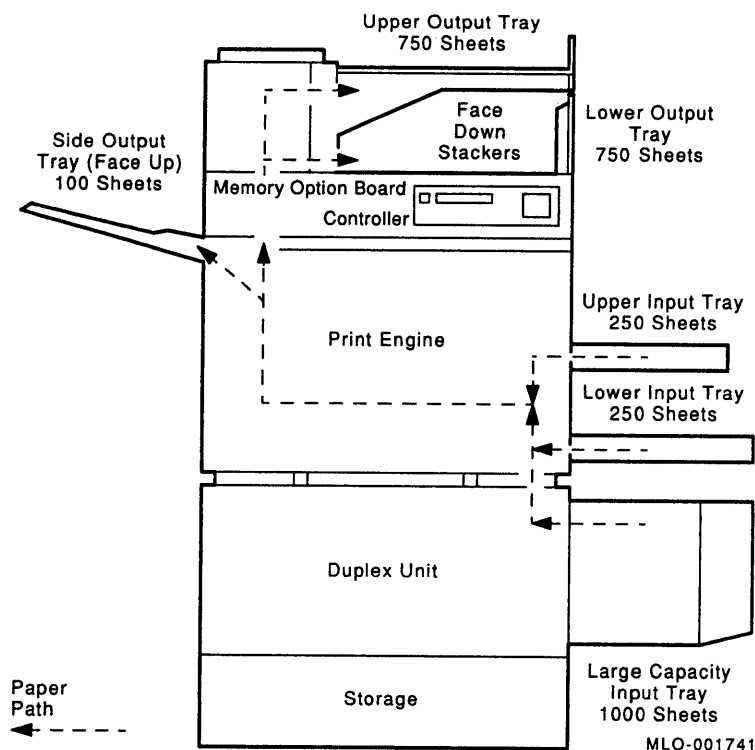
The PrintServer 20 system provides high-quality, shared printing services in an office cluster environment. It supports both *simplex* (one-sided) and *duplex* (two-sided) printing. A two-line liquid crystal display (LCD) on the operator panel displays operator and jam messages.

The printer contains the following hardware components:

- Print engine
- Controller
- Memory option board

Figure 1–5 is an illustration of these components, which are described in the following sections.

Figure 1-5 PrintServer 20 Hardware Components



1.3.1.1 Print Engine

To produce black marks on plain paper, the print engine uses a technique called *laser electrophotography*. In this nonimpact printing technique, a low-power laser generates an image one line at a time on the photoconductive surface of a negatively charged Organic Photo Conductor (OPC) drum. The light discharges the portions of the drum on which the image is to appear. When the printer applies a dry powder (toner) to the drum, the negatively charged toner particles adhere to the discharged areas of the drum. When the drum is pressed against the media and heated, the powder is fused to the media. This imaging technology is called *write-black* printing.

The print engine accepts one-bit/pixel data from bitmap memory and generates images having 300-pixel/inch resolution. Maximum printing speed for the PrintServer 20 print engine is 20 A-size pages per minute in simplex mode. The speed at which data can be printed varies according to the page composition time. Complex pages take longer to process and print.

The print engine has two selectable features:

- Input and output media trays
- Duplex unit

You can select these features either by typing an option to the **lpr** command or by associating the feature with the **/etc/printcap** file for a PrintServer printer.

Input and Output Media Trays

PrintServer printers print either on paper or transparencies. The PrintServer 20 model has two slots for input trays. Each slot can be used for either a small or large media tray. The trays each have a 250-sheet capacity. A large capacity input tray (LCIT) with a 1000-sheet capacity is also available.

The media output devices consist of an upper and a lower face-down offset stacker. Offset stackers separate one job from the next. Each face-down stacker has a 750-sheet capacity. The PrintServer 20 model also has one face-up side output bin with 100-sheet capacity.

Duplex Unit

The print engine on the PrintServer 20 can print in either simplex or duplex mode. That is, it can print on one side or both sides of the sheet.

1.3.1.2 Controller

The PrintServer 20 controller contains a Micro/VAX II processor that links client computers to the print engine through the Ethernet transceiver cable. The controller receives input data in POSTSCRIPT format and changes the data into bitmap images to be passed to the print engine. The controller also executes POSTSCRIPT programs in memory and controls the print engine.

The controller contains the following components:

- **Processors**

The PrintServer 20 controller has two processors that interpret and process instructions in POSTSCRIPT programs and document files: a Micro/VAX II Central Processing Unit (CPU) and a Micro/VAX II Floating Point Unit (FPU).

- **Memory**

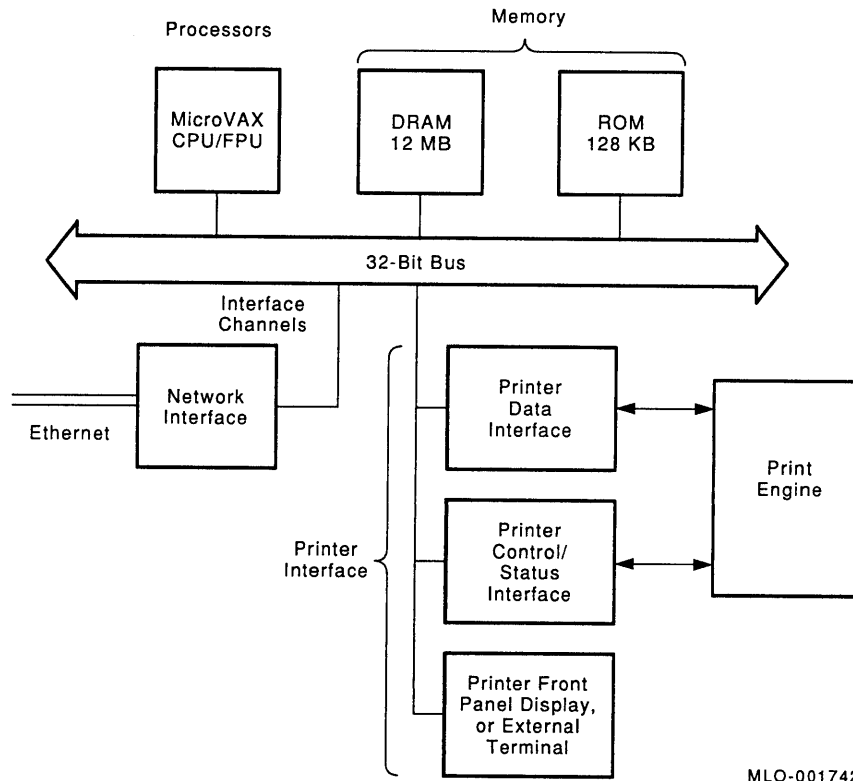
POSTSCRIPT programs are executed in memory. Memory also stores full page bitmaps of page descriptions. The PrintServer 20 controller contains two memory modules: 12 Megabytes Dynamic Random Access Memory (DRAM) and 128 Kilobytes Read-Only Memory (ROM).

- **Interface channels**

Interface channels pass data from the Ethernet network to the controller and from the controller to the print engine. The PrintServer 20 controller has two interface channels: a network interface and a printer interface.

Figure 1-6 is a simplified block diagram of the PrintServer 20 controller.

Figure 1–6 PrintServer 20 Data Controller



1.3.2 The PrintServer 40 Series Hardware

The PrintServer 40 systems support simplex (one-sided) printing.

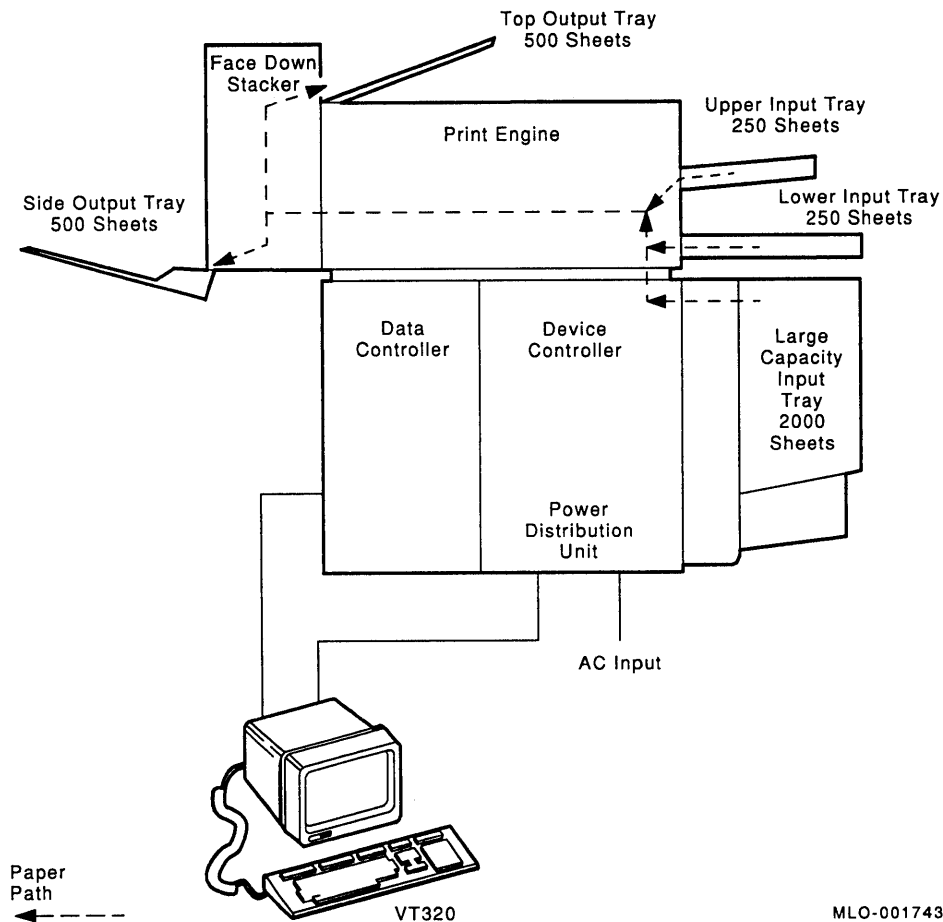
A VT320 terminal is connected to the printer, allowing communication with the PrintServer software. Server managers, who are responsible for maintaining the printer, can read error messages and status information from the terminal screen panel and modify the operating parameters of the PrintServer system.

The PrintServer 40 systems contain the following hardware components:

- Print engine
- Data controller
- Device controller
- VT320 video console
- Ethernet communications device

Figure 1–7 is an illustration of these components, which are described in the following sections.

Figure 1-7 PrintServer 40 Series Hardware Components



MLO-001743

1.3.2.1 Print Engine

Like the PrintServer 20 system, the PrintServer 40 systems print engine uses laser electrophotography to produce black marks on paper. However, the PrintServer 40 systems use an imaging technology called *write-white*. A low-power laser generates an image one line at a time on the photoconductive surface of a positively charged selenium drum. The light discharges those portions of the drum on which no image is to appear. When the printer applies a dry powder (toner) to the drum, the negatively charged toner particles adhere to the positively charged areas of the drum. When the drum is pressed against the media and heated, the powder is fused to the media.

The print engine accepts 1-bit/pixel data from bitmap memory and generates images having 300-pixel/inch resolution. Maximum printing speed is up to 40 A-size page images per minute. The speed at which data can be printed varies according to page composition time; complex pages take longer to process and print.

Input and Output Media Trays

The PrintServer 40 series printers have two slots for input trays. Each slot can be used for either a small or large media tray. The trays have a 250-sheet capacity. There is also a large capacity input tray (LCIT) that has a 2000-sheet capacity. All the input trays support several sheet sizes.

For media output, the PrintServer 40 printers have two 500-sheet capacity output trays: a side tray, which supports face-up and face-down stacking, and a top tray, which supports face-down stacking only.

1.3.2.2 Data Controller

As on the PrintServer 20 system, the PrintServer 40 series controller interprets POSTSCRIPT input data, generates bitmap images of page descriptions to be printed, and controls the print engine. However, the PrintServer 40 series controller functions are handled by individual specialized controllers.

The data controller links the client computers to the print engine. It receives input data in POSTSCRIPT format and changes this data into bitmap information to be passed to the device controller. The data controller contains the following hardware components:

- CPU and system unit
- Image controller
- Ethernet controller

Additionally, the data controller runs software called Printing Services Software. (See Section 1.3.3.)

Figure 1-8 shows how data flows through the data controller subsystem. The processor component contains all the other components, because the other components use the processor. The PrintServer 40 series processor is a Micro/VAX II computer.

Figure 1-8 PrintServer 40 Data Controller

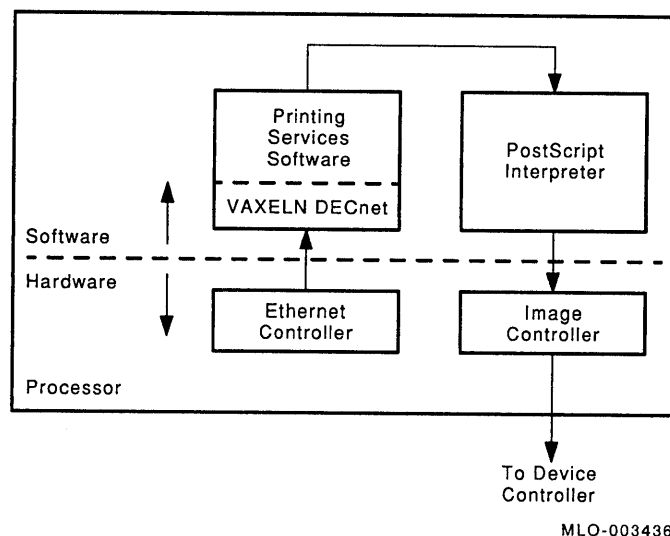


Image Controller

The image controller is a high-performance image generator and controller designed to support page printing for text, graphics, and images. It connects the CPU to the print engine's device controller. The image controller contains processors, memory, and interfaces to receive drawing primitives from the POSTSCRIPT interpreter, to compose images to be printed, and to send bitmaps to the device controller in the print engine.

Drawing primitives are bitmap area operations and character operations created using halftones. The data controller converts the bitmap area operations into bitmap data, and accelerates the transfer of character bitmaps from the CPU to bitmap memory.

Ethernet Controller

The DEQNA module provides the physical interface for the data controller in the PrintServer 40 or PrintServer 40 Plus system to the Ethernet network. The Ethernet controller enables the PrintServer system to communicate with devices connected to the Ethernet coaxial cable. It provides the path to down-line load printing services software from a supporting host.

1.3.2.3 Device Controller

The device controller handles synchronization between the data controller, print engine, and the paper input and output trays. This controller assures that there is proper timing when the paper is fed to the print engine and that the paper is ejected from the engine.

1.3.2.4 Console

A VT320 terminal is connected to the PrintServer 40 or PrintServer 40 Plus system. Server managers can read error messages and status information sent to the terminal screen, and they can modify operating parameters, using a menu.

1.3.3 Printing Services Software

Built on Digital's VAXELN operating system, the Printing Services Software runs on the PrintServer CPU. This software transfers print data from a client; formats the print data, using a built-in POSTSCRIPT interpreter; builds bitmaps; and sends those bitmaps to the data controller subsystem. Printing Services Software is down-line loaded from the supporting host every time a PrintServer printer is booted.

Printing Services Software has four components:

1. A communications protocol provides the communication path for the transfer of the print data from a client to a printer.
2. PrintServer software is the distribution software that accepts print and control data from client systems or a supporting host on the network and passes the data to the imaging software.

The distribution software also returns system messages from the imaging software to the appropriate client or supporting host system.

3. Imaging software interprets data from the distribution software, builds the corresponding bitmaps, and sends them to the print engine to be converted into hard copy.

The imaging software also returns system messages from the print engine or from itself to the distribution software. The PrintServer imaging software contains an interpreter based on the POSTSCRIPT page description language.

4. Management services provide a PrintServer management interface from remote terminals and at the PrintServer 40 or PrintServer 40 Plus console. Management services for the VMS supporting host are described in the *Management Guide: VAX PrintServer Supporting Host*. Management services for the ULTRIX supporting host are described in the *User/Management Guide: PrintServer TCP/IP Supporting Host for ULTRIX*.

For a description of the POSTSCRIPT interpreter, see the *POSTSCRIPT Language Reference Manual* and the *POSTSCRIPT Language Tutorial and Cookbook*.

1.4 Local and Remote Printing

The PrintServer Client for ULTRIX supports both local and remote printing. Traditionally, a local printer was one that was connected to your computer by a wire, and a remote printer was one that was connected to some other computer by a wire. In network printing, the distinction between local and remote printing is less clear.

In the context of PrintServer Client for ULTRIX software, a *local printer* is a printer with which your computer communicates directly, while a *remote printer* is a printer with which your computer does not communicate directly. Instead, your computer sends the file to another computer on the network; that computer in turn communicates with the printer. Operationally, if a printer is local to your computer, then your computer must have the PrintServer Client for ULTRIX software installed on it. If a printer is remote to your computer, your computer must have only a **/etc/printcap** entry for that printer and a spooling directory.

Only client nodes can submit jobs directly to the PrintServer printer. Nonclient nodes (those that are not running the PrintServer Client for ULTRIX software) can submit to remote queues on client nodes on the same network if the access permissions have been set up correctly.

- On a DECnet network, any ULTRIX system can submit jobs to a remote node on the same network. Figure 1–9 shows local and remote printing on a DECnet network.
- On a TCP/IP network, any UNIX system can submit jobs to a remote node on the same network. Figure 1–10 shows local and remote printing on a TCP/IP network.

Figure 1–9 Local and Remote Printing on a DECnet Network

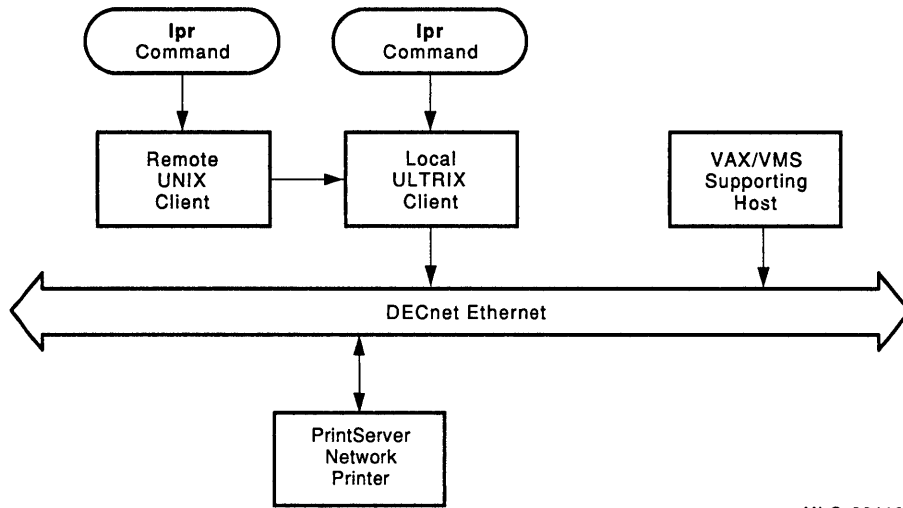
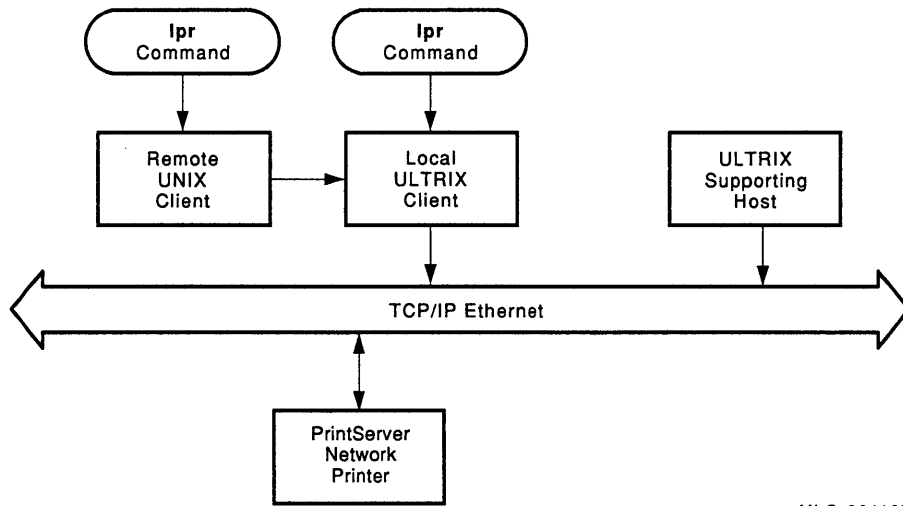


Figure 1–10 Local and Remote Printing on a TCP/IP Network



1.5 PrintServer Configurations

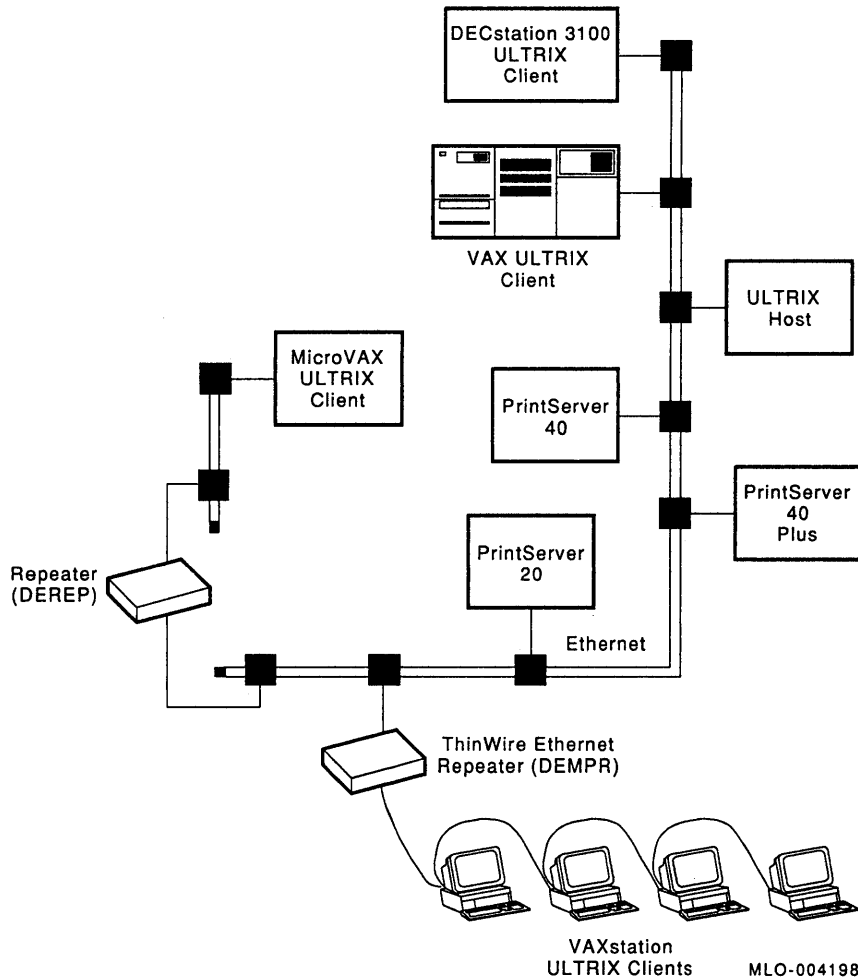
Client software supports a PrintServer configuration for both TCP/IP and DECnet networks. You can print to all supported Digital PrintServer systems in both configurations.

1.5.1 TCP/IP Network

You will need a VAX ULTRIX supporting host to boot and manage PrintServer printers on a TCP/IP network. In this configuration, nonclient systems running on either an ULTRIX or UNIX operating system can spool print files to client systems and print on a PrintServer printer. This allows UNIX and ULTRIX users to share the printing resources of multiple PrintServer systems.

Figure 1-11 is an illustration of PrintServer configuration on a TCP/IP network with an ULTRIX supporting host.

Figure 1-11 Sample TCP/IP PrintServer Configuration

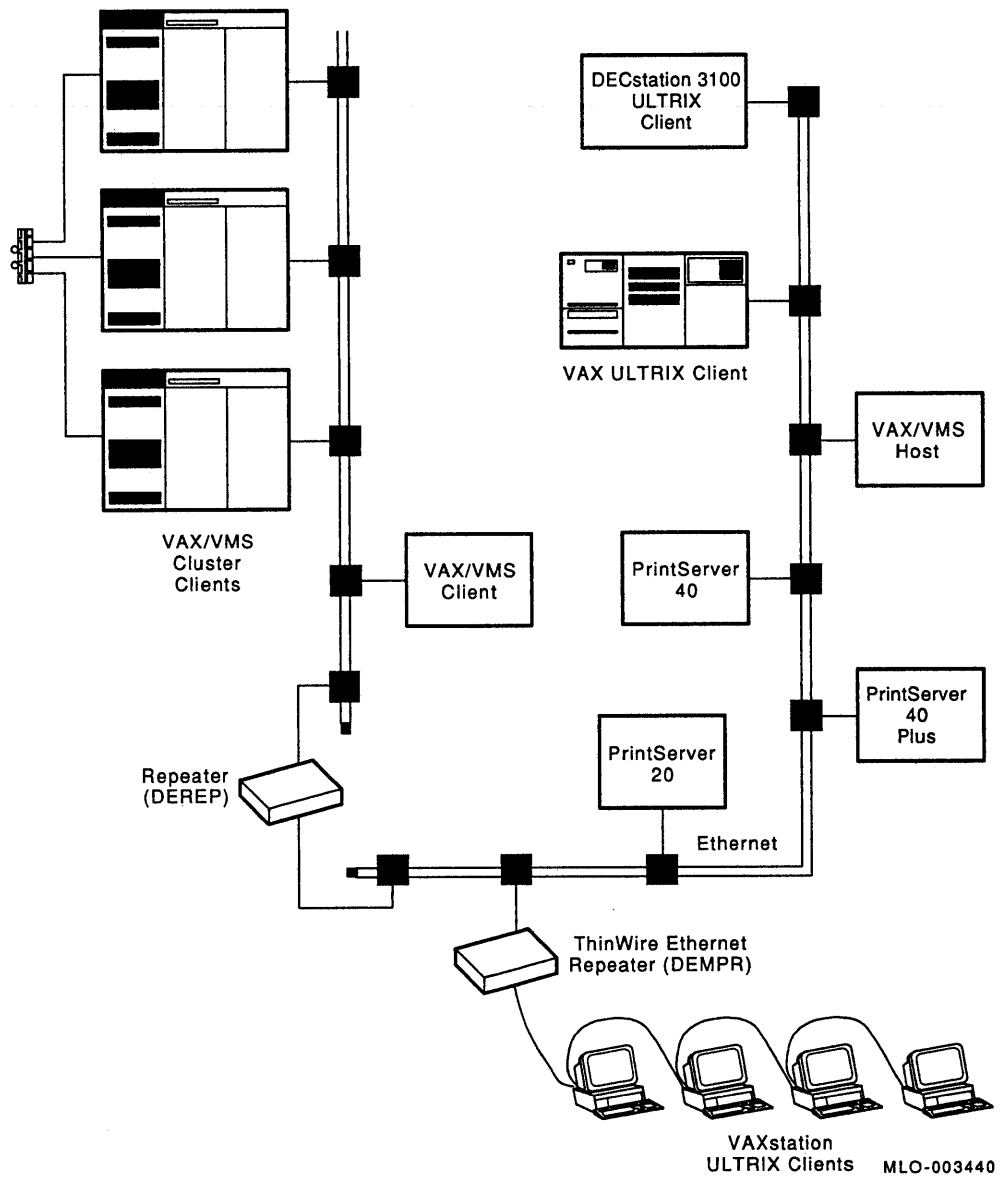


1.5.2 DECnet Network

You will need a VMS supporting host to boot and manage PrintServer printers on a DECnet network. In this configuration, a single VAX node running VMS Supporting Host software can serve both ULTRIX and VMS client nodes. This allows VMS and ULTRIX users to share the printing resources of multiple PrintServer systems.

Figure 1-12 is an illustration of a DECnet PrintServer configuration.

Figure 1-12 Sample DECnet PrintServer Configuration



Printing on a PrintServer System

This chapter provides an overview of printing on a PrintServer system and explains the PrintServer Client for ULTRIX commands that allow you to print files and manage queues. This chapter also describes how to control print jobs with printer setup modules.

2.1 Overview

When powered up or rebooted, the printer requests over the Ethernet network to be loaded with Printing Services Software. The first supporting host that receives the broadcast sends a load file to the printer. Once this code is loaded and started, the printer is ready for printing.

If the PrintServer system boots from a VMS supporting host over a DECnet network, then the client software uses the **lpscomm** communications filter to transfer print data to the printer. If the PrintServer printer boots from an ULTRIX supporting host over a TCP/IP network, then the client software uses the **iplpscomm** communications filter to transfer print data to the printer.

You create print files by running an application program on a client system. These files can be produced by text processing programs, design tools (compilers, CAE/CAD systems), database control programs (spreadsheets), or text and graphic editors.

You issue the PrintServer Client for ULTRIX **lpr** command from a client system, specifying what files to print, when to print them, and the printer control options you want. You must know the data type of the file (POSTSCRIPT, ReGIS, Tektronix 4010/4014, ANSI, or ASCII) and either indicate that data type in the **lpr** command or send the file to a printer that handles that protocol by default. If a translator is necessary, the client software invokes the appropriate translator.

When it is time to print the job, the client attempts to connect to the printer. PrintServer software accepts clients on a first-come, first-served basis. When the printer has handled any clients connected before this request, the printer indicates to the client that it is ready for print data.

The printer receives and processes the data sequentially, building page images and printing them on paper. All fonts referred to by the print file must be contained in the PrintServer software or in the actual print data; otherwise an error occurs.

Depending on the options you specify with the **lpr** command, **lpd** reports errors and job-related information within the job to the operator, the user, or both. The server manager can have events logged on the supporting host. Certain events, such as a paper jam, pause the job until the condition is corrected; some events can cause the job to be terminated.

2.2 lpr Command Options

This section describes the syntax, defaulting scheme, and the function of each **lpr** command option.

The PrintServer Client for ULTRIX expands the ULTRIX **lpr** command options to allow you to use the printing features of Digital's family of network printers. This section describes the PrintServer options. For more information on **lpr(1)**, see the *ULTRIX Reference Pages*.

2.2.1 Syntax

The syntax of the ULTRIX **lpr** command is:

```
lpr [-flag[qualifier]] [...] [filename...]
```

In this syntax, *flag* is a single, case-sensitive letter or number used to select options; *qualifier* is a number or text string that qualifies or further describes the selected option; *filename* is zero or more files to which the **lpr** command is applied. If no files are specified, input is taken from the standard input.

You can enter flags in any order. Some options are mutually exclusive and some work in combination with others. These are noted in the description of the options in this section. The flags apply equally to all files in the file list. Positional options are not supported. In other words, a flag cannot apply to selective files in a multifile list.

If you enter conflicting or duplicate options on the command line, the last option entered is the one that is used.

The following rules apply to **lpr** options on a PrintServer system:

- The first option flag must be preceded by a dash (-).
- Multiple unqualified options can either be abutted or separated by white space and a dash (-).

```
% lpr -h -s filename  
% lpr -hs filename
```
- Spaces are optional between an option flag and a qualifying string.

```
% lpr -P postscript filename  
% lpr -Ppostscript filename
```
- Multiple qualified options must be separated by a space, and each qualifier must be separated by a dash.

```
% lpr -Ppostscript -JReport filename
```
- Options that require a qualifier must have a qualifier supplied. The following example causes the string *-h* to be interpreted as a printer name.

```
% lpr -P -h
```

Table 2-1 lists the **lpr** option flags and qualifiers that select PrintServer functions. Boldface characters indicate the minimum characters required to enter a value for a qualifier.

NOTE

In the PrintServer Client for ULTRIX, the **-i** option to the **lpr** command requires a numeric qualifier. This option is not compatible with the ULTRIX **lpr** command, in which the qualifier is optional.

Table 2–1 lpr Options for PrintServer Printers

Option	Flag	Qualifiers
data_type	-D	ansi , ascii , postscript , regis , tek4014
input_tray	-I	top , middle , bottom , lcit
layup_definition	-L	A filename
message	-M	ignore , keep
number_up	-N	A number from 0 through 100
orientation	-O	portrait , landscape
output_tray	-o	top , side , face-up , upper , lower , lcos
page_limit	-Z	[lowlim] [,uplim] [1–10000]
page_size	-F	letter or a , a3 , a4 , a5 , ledger or b , b4 , b5 , executive , legal
sheet_count	-X	A number from 1 through 10,000
sheet_size	-S	letter or a , a3 , a4 , a5 , ledger or b , b4 , b5 , executive , legal
sides	-K	1 or one_sided_simplex , 2 or two_sided_duplex , tumble or two_sided_tumble , one_sided_duplex , one_sided_tumble , two_sided_simplex

2.2.2 Defaulting Scheme

If you do not enter an option on the **lpr** command line, **lpd** follows this general defaulting scheme to determine the option value. (Specific default values are given in the discussion of each **lpr** option.)

1. Interaction with other **lpr** options. Options such as **page_size** and **sheet_size** are related. If you specify one option, and not its related option, the value you specify is used for both options.
2. Entry in the **/etc/printcap** file. You can specify a default value for some **lpr** options in the printer description in the **/etc/printcap** file. Chapter 3, Configuring the Client, in the *Installation Guide: PrintServer Client for ULTRIX* explains how to specify default values in the **/etc/printcap** file.
3. Line printer daemon defaults. For example, if no **data_type** option is specified on the **lpr** command line or in the **/etc/printcap** file, **lpd** assumes the **data_type** is **POSTSCRIPT**.
4. PrintServer defaults. For example, a particular PrintServer printer can have a default input and output media tray.

2.2.3 data_type

-Ddata_type

The **-D** option flag determines if the data to be printed must be translated into the **POSTSCRIPT** language. **POSTSCRIPT** data is sent to the PrintServer printer without translation. If the data is other than **POSTSCRIPT**, the print daemon calls the appropriate data syntax translator to convert the data before sending it to the printer.

If **-D** is not specified, the default `data_type` is found in the `/etc/printcap` file. If no `data_type` entry exists in the `/etc/printcap` file, the print job is sent to the printer without translation.

Table 2–2 lists the qualifiers to the `data_type` option.

Table 2–2 data_type Qualifiers

Qualifier	Description
ansi	ANSI data
ascii	ASCII data
postscript	POSTSCRIPT data
regis	ReGIS data
tek4014	Tektronix 4010/4014 data

For more information on ANSI, ReGIS, and Tektronix 4010/4014 data, see the *POSTSCRIPT Translators Reference Manual*. For more information about POSTSCRIPT data, see the *Programmer's Supplement: POSTSCRIPT Printers*.

ASCII data is generated by **ln03rof**, an LN03R ASCII-to-POSTSCRIPT translation filter. The **ln03rof** translation filter prints a job in *landscape* mode with a slightly smaller font if the width value exceeds 80 characters. If the length value exceeds 88 lines, a smaller font is used. For more information on **ln03rof(4)**, see the **man(1)** pages and the *ULTRIX Reference Pages*.

The `data_type` option is offered in addition to other ULTRIX filters that are already available, such as **-n** (the **ditroff** filter) and **-p** (the **pr** filter).

Examples

1. This example translates an ANSI file to POSTSCRIPT before printing it on a PrintServer printer:

```
% lpr -P fast -Dansi filename
```
2. This example sends the file directly to a Printserver printer, with no translation:

```
% lpr -P hi_speed filename_ps
```

2.2.4 input_tray

-Iinput_tray

The **-I** option flag selects the input tray that will supply the paper for the job.

Table 2–3 lists the qualifiers to the `input_tray` option and the selected tray on the PrintServer 40 series and PrintServer 20 models.

Table 2–3 PrintServer Input Trays

Qualifier	PrintServer 40 Series	PrintServer 20
top	Upper 250 sheet	Upper 250 sheet
middle	Lower 250 sheet	Lower 250 sheet

(continued on next page)

Table 2–3 (Cont.) PrintServer Input Trays

Qualifier	PrintServer 40 Series	PrintServer 20
bottom or LCIT	Large capacity input tray	Large capacity input tray

The **-I** option is used with `sheet_size` (**-S**) to select an input tray. If both **-I** and **-S** are specified, the selected input tray must contain the required sheet size. If neither **-I**, **-S**, nor **-F** (page size) is specified, the **-I** values are taken from the `/etc/printcap` file. If these options are not specified in the `/etc/printcap` file, the default paper tray for the printer is used.

2.2.5 layup_definition

-L*layup_definition*

The **-L** option flag includes a layup definition file in your print job. The layup definition file sets all of the *layup* features, except number up. (Number up refers to printing multiple pages on a sheet of paper. See Section 2.2.7.) Examples of layup features include setting a page grid, alternating margins, and drawing page borders. The layup definition file is translated to POSTSCRIPT before your job is printed.

This section explains how include a layup definition file in a print job. See Chapter 3 for information on creating a layup definition file.

- You can call a layup definition file either by specifying it on the **lpr** command line or in the `/etc/printcap` file. If the layup file name does not begin with `/`, the current directory is searched, followed by `/usr/lib/lpddfilters`. If you submit a print job from a remote print queue, the layup definition file must exist on the local node.
- If you specify a layup definition file without specifying `number_up`, the default `number_up` value is 1, even if `number_up` is specified in the `/etc/printcap` file. Similarly, if you specify `number_up` without specifying a layup definition file, your print job does not use a layup definition file, even if a layup value is specified in the `/etc/printcap` file. So, to use a layup definition file and `number_up` together, you must specify both options either on the command line or in the `/etc/printcap` file.
- If you specify both a layup definition file and `number_up` and the value for `number_up` is 0, the job will print without the layup definition file.
- Four layup definition files are included as part of the client software in the directory `/usr/lib/lpddfilters`. (See Chapter 3 for a description of these files.)
 - **lpsnup.lup** sets margins for number-up printing.
 - **lpssingleholes.lup** sets a wide left margin for hole-punching pages for single-sided print jobs.
 - **lpdoubleholes.lup** sets a wide left margin for hole-punching on alternate pages for two-sided printing.
 - **lpsholes.lup** sets a wide left margin for hole-punching without scaling down the size of the page image.

Example

This example uses the layup definition file **lpssingleholes.lup**, which sets up a print job with a wide left margin for hole-punching single-sided pages. This print job uses the default `number_up` value 1, which prints one page per sheet, on a PrintServer system called *fast*.

```
% lpr -P fast -Llpssingleholes.lup filename
```

2.2.6 message

-Mmessage

The **-M** option flag specifies what happens to job-generated event messages and user data returned by POSTSCRIPT. Table 2–4 lists the qualifiers to the message option.

Table 2–4 message Qualifiers

Qualifier	Description
keep	Messages are mailed to the user.
ignore	Messages are not recorded, and the message file is not created. This is the default.

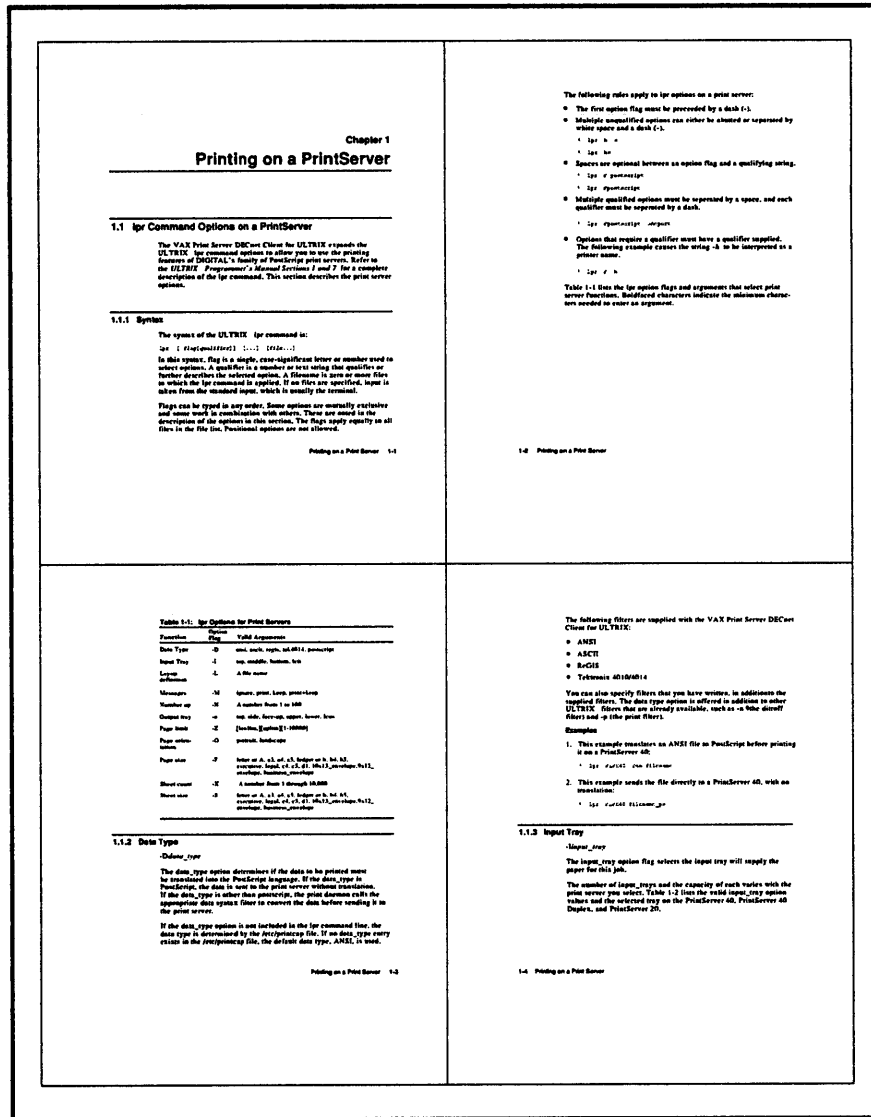
The default is that messages are ignored, unless a message option is specified in the **/etc/printcap** file. Chapter 3, Configuring the Client, in *Installation Guide: PrintServer Client for ULTRIX*, explains how to make this entry in the **/etc/printcap** file.

2.2.7 number_up

-Nnum

The graphic arts term for printing multiple pages on a sheet is *number up*. (A sheet is a piece of paper. A page is an image, usually printed on a single sheet.) The **-N** option flag sets the maximum number of *page spots* on a sheet. These are the positions on a sheet where a page could be printed. For example, if `number_up` is 4 (**-N4**), you can print four pages on a single sheet. This means there are four page spots on the sheet, even if your print job actually has fewer than four pages. Figure 2–1 shows an example of a sheet printed with a `number_up` value of 4.

Figure 2-1 Four Pages to a Sheet (4-Up)



MLO-004192

Your job prints using the number set by the **-N** option, unless you use a layup definition file that contains the **PagesPerSheet** option or the **Grid** option. The default is to print one page on a sheet (**-N1**), but you can set a default `number_up` value in the `/etc/printcap` file.

`num` can have a value from 0 to 100.

To disable layup, specify **-N0**. This overrides any layup defaults that are set in the `/etc/printcap` file.

2.2.7.1 `number_up` Option and Page Orientation

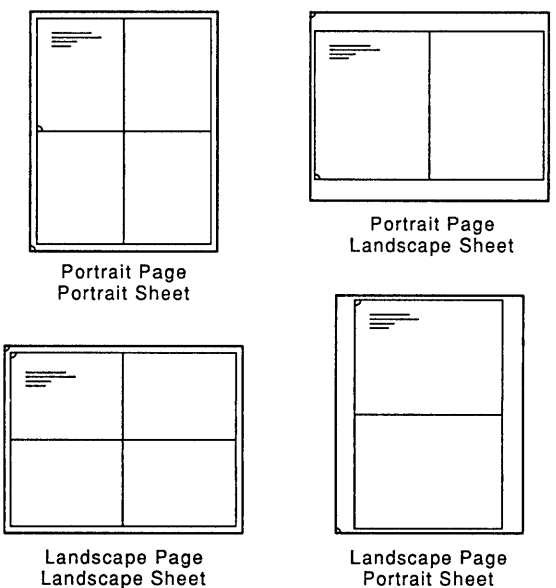
When you print using the `number_up` option, the *aspect ratio* of the page is preserved. This means that, for example, a portrait-shaped page retains its *portrait* shape. However, the page and sheet may have different orientations. For example, if you print portrait pages 2-up (**-N2**), the pages remain portrait-shaped, but you hold the sheet in landscape orientation to read them.

If you specify that the pages are to print as landscape pages, you hold the sheets in portrait orientation to read them.

Figure 2–2 shows the four different cases of page and sheet orientation with the **-N** option. Text in the first page spot is represented by the parallel lines. The POSTSCRIPT origin of the sheet and first page are represented by the quarter circles.

See Section 2.2.8 for a description of the page orientation option (**-O**).

Figure 2–2 Page Orientation with `number_up`



MLO-001476

2.2.7.2 `number_up` Option Paper Selection Options

The **-N** option sizes the page image according to the page size in effect at the time the print job is submitted.

If you specify **-N1** for a file that uses a POSTSCRIPT tray selection operator, the printer changes paper trays, but the image is not resized to fit the new paper.

If you specify **-N** for a number greater than one, POSTSCRIPT tray selection operators in the file are ignored. Because ANSI files are translated to POSTSCRIPT before printing, ANSI tray selection sequences are also ignored when ANSI files are printed using **-N**.

If you want to print on paper other than the paper in the default input tray when you are using the `number_up` option, use the **-I** (`input_tray`) and **-F** (`page_size`) options.

Examples

1. This example prints the file eight pages to a sheet:

```
% lpr -P fast -N8 filename
```
2. This example prints the file eight pages to a sheet and uses a layup definition file:

```
% lpr -P fast -Llpssingleholes.lup -N8 filename
```
3. This example prints the file 2-up (two pages to a sheet) in landscape orientation. Note that the sheet and the pages do not always have the same orientation (see Section 2.2.8.1):

```
% lpr -P fast -Olandscape -N2 filename
```
4. This example prints the file six pages to a sheet. It prints on 11 x 17-inch paper, which is loaded in the middle input tray on this printer:

```
% lpr -P bigprinter -Imiddle -N6 filename
```

2.2.8 orientation

-O*orientation*

The **-O** option flag specifies the orientation of the printed output on the logical page.

Table 2–5 lists the qualifiers to the orientation option:

Table 2–5 orientation Qualifiers

Qualifier	Description
portrait	The lines of text are oriented parallel to the short side of the logical page. (This page is printed in portrait mode.)
landscape	The lines of text are oriented parallel to the long side of the logical page.

The **-O** option works only if the ANSI, Tektronix 4010/4014, or ReGIS translator is used. POSTSCRIPT files ignore the page orientation set by the **-O** option, because attributes like page size and orientation are set in the file itself. If you want to change the page orientation, you have to edit the POSTSCRIPT file.

Example 2–1 shows POSTSCRIPT code you can add to the top of a file to change the orientation from portrait to landscape for an 8.5 x 11-inch page. The number 612 represents the number of *points* in 8.5 inches. (There are 72 points to an inch.) The number 90 represents the number of degrees the page image is rotated.

Example 2-1 Sample POSTSCRIPT Code to Change from Portrait to Landscape Orientation

```
612 0 translate
90 rotate
```

2.2.8.1 orientation with number_up

The **-O** option affects the page, not the sheet. When you print using the **-N** option with the **-O** option, the page orientation and the sheet orientation are not always the same. For example, if you print portrait pages 2-up, the pages remain portrait-shaped, but you may hold the sheets in landscape orientation to read them. If you change the page orientation to landscape, the pages are printed as landscape pages, but you hold the sheets in portrait orientation to read them. (See Figure 2-3.)

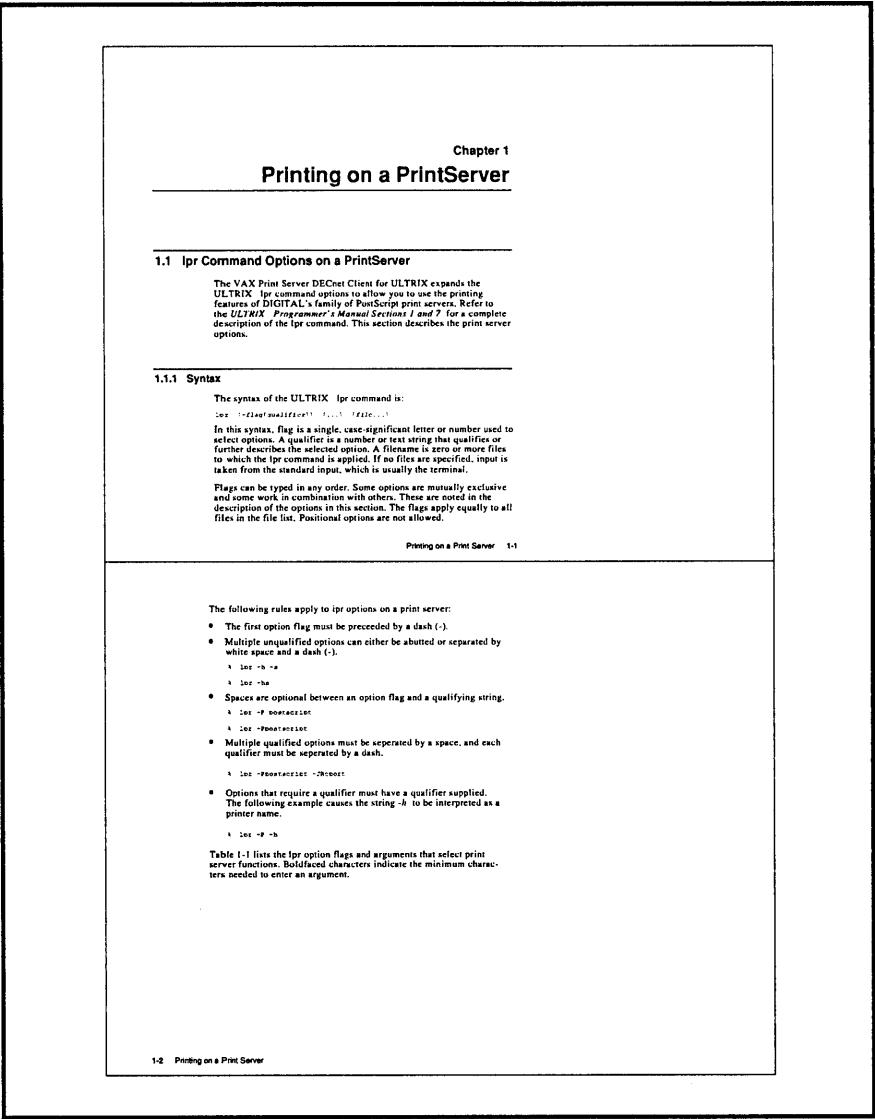
2.2.9 output_tray

-o*output_tray*

The **-o** option flag selects the tray in which the PrintServer printer deposits the printed output.

Table 2-6 lists the qualifiers to the *output_tray* option.

Figure 2-3 Landscape Pages on a Portrait Sheet



MLO-004191

Table 2–6 output_tray Qualifiers

Qualifier	Description
face_up	Side tray, with face-up stacking
lcos	Large capacity output stacker
lower	Lower tray, if there are two trays on top of the printer. If there are not two trays, the top tray is used.
side	Side tray, with face-down stacking
top	Top tray, with face-down stacking
upper	Upper tray, if there are two trays on top of the printer. If there are not two trays, the top tray is used.

The output trays on each printer can vary as to number, sheet capacity, and type of stacking (face up or face down).

If you choose an output tray that does not exist on a PrintServer network printer, an appropriate tray is substituted. In some cases, one tray is substituted for another because the two perform the same function, even though they have different locations on different printers.

For example, specifying the side output tray (**-oside**) on the PrintServer 40 models produces face-down output. The side output tray on the PrintServer 20 model does not provide face-down stacking. Therefore, the lower output tray (which provides face-down stacking) is substituted for the side output tray on the PrintServer 20 printer. The reverse is also true; if you specify the lower output tray (**-olower**), output will go to the side tray on the PrintServer 40 printers.

Table 2–7 lists the valid output_tray qualifiers and the selected tray on each PrintServer printer.

Table 2–7 Valid output_tray Qualifiers on PrintServer Models

Qualifier	PrintServer 20	PrintServer 40 Series
face_up	face_up	face_up
lcos	top	lcos
lower	lower	side
side	lower	side
top	upper	top
upper	upper	top

If you do not specify a value for the output_tray option, the default value in the **/etc/printcap** file is used. If no value is specified in the **/etc/printcap** file, the selected printer determines the default.

2.2.10 page_limit

-Z[*lo_page_lim*] [*hi_page_lim*]

The **-Z** option flag specifies the numbers of the first and last pages to be printed for the job. You can specify any value from 1 through 10,000.

The following points explain the operation of the *page_limit* option:

- If this option is omitted, the entire job is printed.
- If the lower limit is omitted, printing starts on the first page of the job.
- If the upper limit is omitted, printing continues to the end of the job.
- The upper and lower limits apply to the entire job and not to each file of a multifile job. Therefore, if you intend to print a selected portion of a file, issue a separate print request for that file alone.
- When you use the *page_limit* option (**-Z**), your print job aborts when the upper limit is reached. This means that you should not use the *page_limit* option with the *copies* option (**-#**), because the print job may abort before the second copy begins printing. If you want to print multiple copies of a job for which you specify the page limit, use the *sheet_count* option.

Examples

1. This example prints the first ten pages of a job:

```
% lpr -P fast -Z,10 filename
```

2. These examples both print from page 5 to the end of the file:

```
% lpr -P fast -Z5, filename
```

```
% lpr -P fast -Z5 filename
```

The comma is optional when you specify a lower limit and want to print to the end of the file.

3. This example prints pages 10 through 13 on one sheet of 11 x 17-inch paper:

```
% lpr -P big -N4 -Z10,13 -Sb filename
```

4. This example prints two uncollated copies of the first three pages of a job:

```
% lpr -P hi_speed -X2 -Z,3 filename
```

2.2.11 page_size

-F*page_size*

The **-F** option flag specifies the size of the logical pages being printed. When used with the **-S** (sheet-size) option, the **-F** option determines the scale factor and aspect ratio of the logical pages.

This option works only if you use the ANSI, Tektronix 4010/4014, or ReGIS translator.

Table 2–8 lists the qualifiers to the *page_size* option.

Table 2–8 page_size Qualifiers

Qualifier	Page Size
letter or a	8.5 x 11 in (216 x 279 mm)
ledger or b	11 x 17 in (279 x 432 mm)
legal	8.5 x 14 in (216 x 356 mm)
executive	7.5 x 10.5 in (191 x 267 mm)
a5	5.8 x 8.3 in (148 x 210 mm)
a4	8.3 x 11.7 in (210 x 297 mm)
a3	11.7 x 16.5 in (297 x 420 mm)
b5	7.2 x 10.1 in (176 x 250 mm)
b4	10.1 x 14.3 in (257 x 264 mm)

If you do not specify **-F**, the sheet size value determines the page size value. If you do not specify **-F** or **-S**, the page size is determined by the **/etc/printcap** file. If no page size entry exists in the **/etc/printcap** file, the default page size is letter (or **a**).

2.2.12 sheet_count

-Xnum

The **-X** option flag specifies the number of times each sheet is to be printed. This option gives you uncollated output: that is, copies of the first sheet, followed by copies of the second sheet, and so on.

num can be any number from 1 through 10,000. The default *sheet_count* value is 1.

If you need collated output, use the **-#** (copies) option flag in the **lpr** command. The **-#** option flag gives you sets of consecutive sheets.

NOTE

The **-#** option flag takes more time to process and increases network traffic because the client must send multiple copies of each bitmap.

2.2.13 sheet_size

-Ssheet_size

The **-S** option flag specifies the size of the sheets being printed.

Table 2–9 lists the qualifiers to the **-S** option flag.

Table 2–9 sheet_size Qualifiers

Qualifier	Sheet Size
letter or a	8.5 x 11 in (216 x 279 mm)
ledger or b	11 x 17 in (279 x 432 mm)

(continued on next page)

Table 2–9 (Cont.) sheet_size Qualifiers

Qualifier	Sheet Size
legal	8.5 x 14 in (216 x 356 mm)
executive	7.5 x 10.5 in (191 x 267 mm)
a5	5.8 x 8.3 in (148 x 210 mm)
a4	8.3 x 11.7 in (210 x 297 mm)
a3	11.7 x 16.5 in (297 x 420 mm)
b5	7.2 x 10.1 in (176 x 250 mm)
b4	10.1 x 14.3 in (257 x 264 mm)

The following points explain the operation of the `sheet_size` option:

- If you specify a sheet size that is not available on your PrintServer system, your job does not print, and, if you have specified **-Mkeep**, a message is mailed to you with an explanation of the problem.
- If you do not specify the **-S** option, the page size value determines the sheet size. If you also do not specify page size (**-F**), the sheet size is determined by the sheet size value in the `/etc/printcap` file. If no entry for sheet size exists in the `/etc/printcap` file, the default sheet size is letter (or **a**).
- If you use both the input tray (**-I**) and sheet size (**-S**) options, the input tray you specify must contain the sheet size you specify; otherwise your job does not print and a message is returned with an explanation of the problem.

2.2.14 sides

-Ksides

The **-K** option flag determines whether pages are printed on one or both sides of a sheet and, in some cases, can select whether the layout of pages is simplex, duplex, or *tumble* printing.

The `sides` option can select the page layout only if the application you used to create your document allows you to select the layout at the time of printing. If the application does not permit this, the layout information for the option is ignored. In this case, the option selects whether the pages are printed on one side or both sides of the sheet and whether or not they are printed tumble.

NOTE

You cannot print files using the `sides` option from a PrintServer Client for ULTRIX system if the supporting host is running PrintServer software at Version 2.1 or earlier.

Terminology

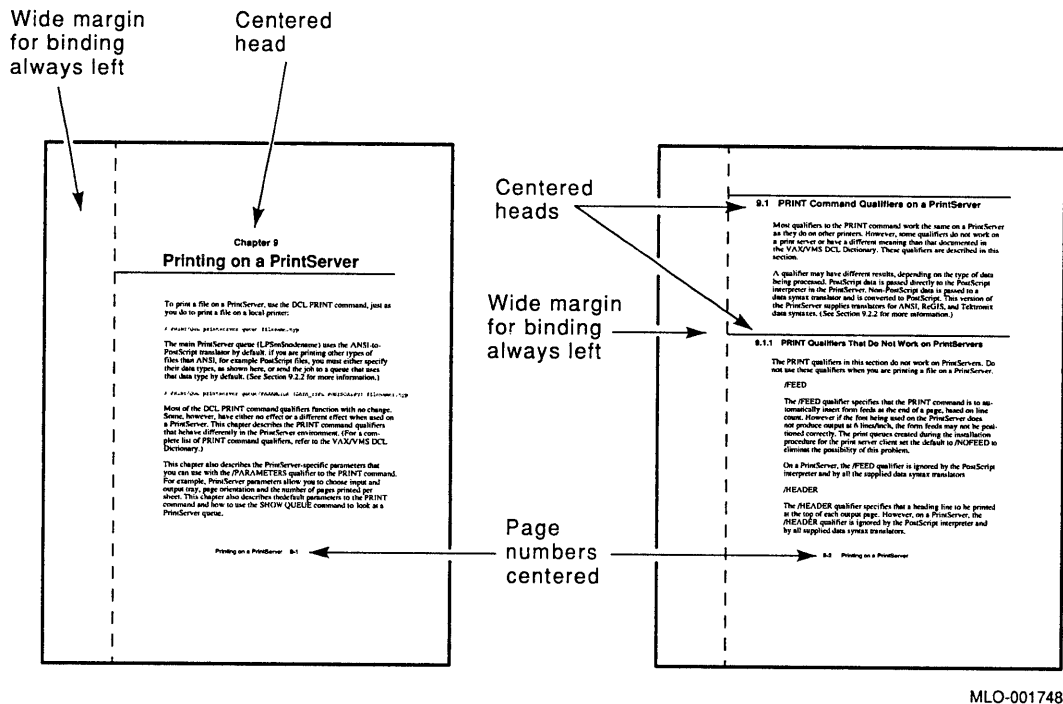
To understand the values you can give to `sides`, you need to understand the following terms:

recto In a book, recto pages are odd-numbered pages. In an open book, the recto page is on the right. For example, page 1 is always a recto page. For a single sheet, the recto page is the page on the front of the sheet.

verso In a book, verso pages are even-numbered pages. In an open book, the verso page is on the left. For example, page 2 is always a verso page. For a single sheet, the verso page is the page on the back of the sheet.

simplex layout Documents in simplex layout, sometimes called logical simplex, are formatted to be printed on one side of a sheet. The page headings and page numbers appear in the same position on each page. The margins remain the same for every page. For example, if the document is to be put in a binder, every page will have a wide left margin for hole-punching. (See Figure 2-4.)

Figure 2-4 Sample Layout for One-Sided Printing



MLO-001748

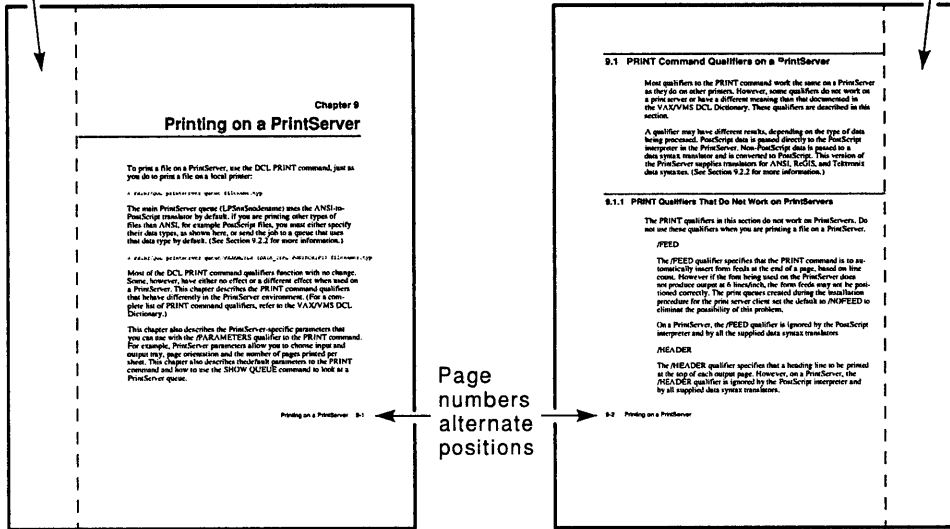
simplex printing Simplex printing, sometimes called physical simplex, refers to printing on one side of a sheet.

duplex layout Documents in duplex layout, sometimes called logical duplex, are formatted to be printed on two sides of a sheet. Page headings and page numbers often alternate right and left, so that when the sheet is printed they always appear on the outside edge of the page. In a document that has chapters, the first page of each chapter is a recto page, so duplex-formatted documents sometimes have a blank page inserted as the last page of a chapter if the chapter ends on a right-hand page. Duplex documents are to be bound along the long (usually the left) edge of the sheet. The margin that remains for binding or hole-punching alternates, rather than remaining on the same side of the page. (See Figure 2-5.)

Figure 2-5 Sample Layout for Two-Sided Printing

Wide margin
for binding
alternates
right and left

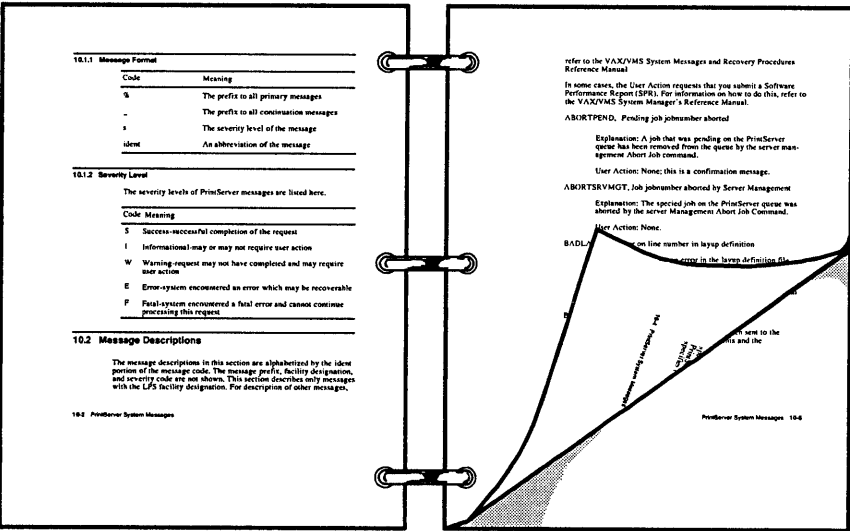
Wide margin
for binding
alternates
right and left



MLO-001749

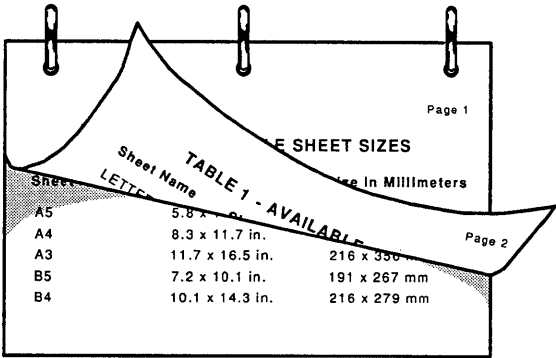
duplex printing Duplex printing, sometimes called physical duplex, refers to printing on both sides of a sheet. Documents printed in normal duplex, rather than tumble duplex, are to be bound along the long (usually left) edge of the sheet. (See Figure 2-6 and Figure 2-7).

Figure 2-6 Sample of Two-Sided Printing (Portrait Page)



MLO-001750

Figure 2-7 Sample of Two-Sided Printing (Landscape Page)



MLO-001853

tumble layout Tumble layout is for two-sided documents that are to be bound along the short edge of the sheet.

For example, a tumble document may be bound at the top of each sheet, so you turn each page from the bottom. Legal documents are often tumble printed.

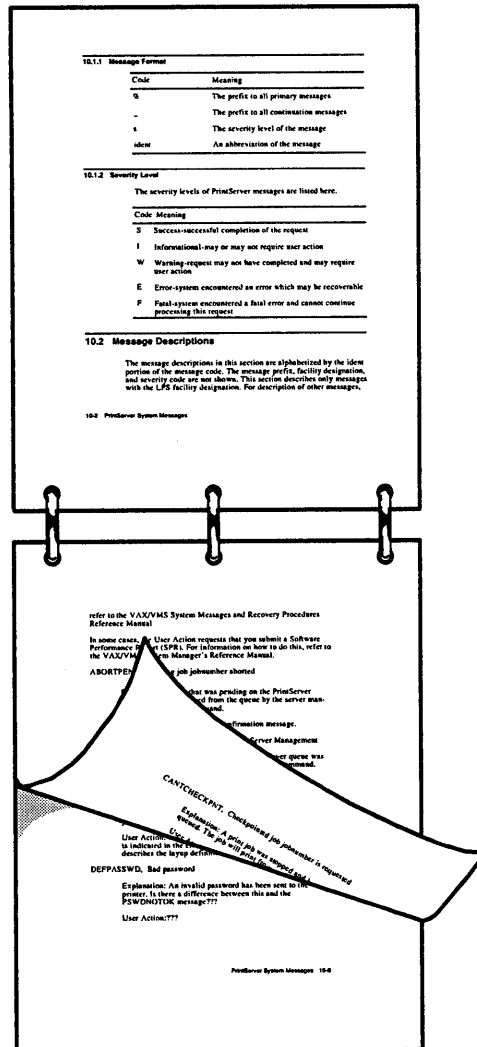
As with duplex format, the page headings may alternate position. However, in tumble format, rather than alternating right and left, recto pages may have the page heading at the bottom of the page, and verso pages may have the page heading at the top.

As with duplex format, the margin that remains for binding or hole-punching alternates, but it alternates along the two short edges of the sheet, rather than along the two long edges.

As with duplex format, recto pages are printed on the first side of a sheet, so blank pages may be inserted where necessary.

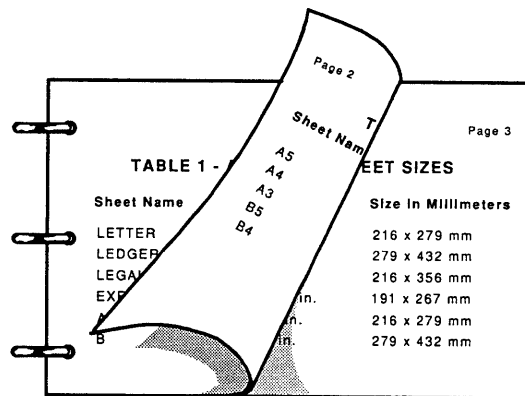
tumble printing In tumble printing alternate pages are rotated 180 degrees. You can print using one-sided or two-sided tumble. For examples of tumble printing see Figure 2-8 and Figure 2-9.

Figure 2-8 Sample of Tumble Printing (Portrait Page)



MLO-001751

Figure 2-9 Sample of Tumble Printing (Landscape Page)



MLO-001852

Errors with sides Option

Not all printers support two-sided printing. If you specify two-sided printing on a printer that does not support that feature, your job is aborted and an error message is displayed.

Table 2-10 describes the valid qualifiers to the sides (-K) option flag.

Table 2-10 sides Option Qualifiers

Qualifier	Description
1 or one_sided_ simplex	Output is printed on one side of each sheet. If the application you used to format your document allows you to select a format at print time, this value lets you select simplex format.
2 or two_sided_ duplex	Output is printed on both sides of each sheet. Logical pages are oriented to physical sheets in the same way on all sheets. The printing in this book is an example of two-sided duplex printing. If the application you used to format your document allows you to select a format at print time, this value lets you select duplex format.
tumble or two_sided_ tumble	Output is printed on both sides of each sheet. Logical pages are oriented 180 degrees in opposite directions on alternating sheets, so printing appears rightside up on one side of the sheet, and upside down on the other side of the sheet. Two-sided tumble printing is useful for printing documents that will be top bound and can be read by flipping the sheets along the top axis. Even though the application you used to format your document did not allow for tumble formatting, specifying tumble or two_sided_tumble always rotates alternate pages 180 degrees.
one_sided_duplex	Your job uses duplex format, but prints on only one side of the sheet. This allows you to make one duplex-formatted copy to take to a copying facility for multiple copies.
one_sided_tumble	Your job uses tumble format, but prints on only one side of the sheet. This allows you to make one tumble-formatted copy to take to a copying facility for multiple copies.

(continued on next page)

Table 2–10 (Cont.) sides Option Qualifiers

Qualifier	Description
two_sided_simplex	Your job uses simplex format, but prints on both sides of the sheet. This allows you to use the fewest possible number of sheets, because it does not insert blank pages in the document.

Examples

1. This example prints an ANSI job on both sides of a sheet.
`% lpr -K2 -Pfast -Dansl file`
2. This example prints a POSTSCRIPT job in tumble mode.
`% lpr -Ktu -Pfast file`
3. This example prints an ASCII job on one side on the sheet, and rotates alternate sheets 180 degrees.
`% lpr -Kone_sided_tumble -Dasci file`

2.2.14.1 sides Option with Page Orientation

Documents printed using normal two-sided printing are meant to be bound along the long edge of the sheet. Documents printed using tumble printing are meant to be bound along the short edge of the sheet. Page orientation does not affect how the sheets are meant to be bound.

If you print a landscape page using normal two-sided printing, you hold the page along the top (long) edge of the sheet and turn it up to read the other side. (See Figure 2–7.)

If you print a landscape page using tumble two-sided printing, you hold the page along the left (short) edge of the sheet and turn it as you would a page of a book to read the other side. (See Figure 2–9.)

2.2.15 Error Messages

This section lists the **lpr** command error messages. The **lpr** command checks the syntax and value limits of all command options. Errors that occur on a remote system are reported by mail.

cannot access file 'file'

Explanation: The specified layout definition file is not accessible.

Action: Check the file name and/or file protection.

cannot open printer description file

Explanation: The `/etc/printcap` file is not accessible.

Action: Check with the system administrator.

illegal option - - 'option'

```
Usage: lpr [-Pprinter] [-#num] [-Cclass] [-Jjob] [-Ttitle]
          [-inum] [-1234font] [-wnum] [-znum] [-Ddata_type]
          [-Iinput_tray] [-Ooutput_tray] [-Oorientation]
          [-Fpage_size] [-Z{lo_page_lim} [,hi_page_lim]
          [-Xsheet_count] [-Ssheet_size] [-Mmessage]
          [-Nnumber_up] [-Llayup_definition] [-Ksides]
          [-cdfghlmnpqrstvx] [filename...]
```

Explanation: You have entered an option that is not supported by the **lpr** command.

Action: Check the spelling of the option. Enter an option supported by the **lpr** command.

invalid argument 'argument'

Explanation: The argument is out of the allowed range.

Action: Enter a value within the accepted range. Refer to the section in this chapter that discusses the option.

invalid argument 'number' - must be in range 1-100

Explanation: You have specified a number that is outside the range 1 to 100 with the **number_up (-N)** option.

Action: Select a number from 1 to 100 as the value for the **number_up (-N)** option.

multiple copies are not allowed

Explanation: An entry in the **/etc/printcap** file restricts the number of copies you can print.

Action: Ask the system administrator to change the entry in the **/etc/printcap** file or choose another printer set up to print multiple copies.

only 'number' copies are allowed

Explanation: An entry in the **/etc/printcap** file restricts the number of copies.

Action: Ask the system administrator to change the entry in the **/etc/printcap** file or choose another printer to print the number of copies you require.

option requires an argument - - 'option'

```
Usage: lpr [-Pprinter] [-#num] [-Cclass] [-Jjob] [-Ttitle]
          [-inum] [-1234font] [-wnum] [-znum] [-Ddata_type]
          [-Iinput_tray] [-Ooutput_tray] [-Oorientation]
          [-Fpage_size] [-Z{lo_page_lim} [,hi_page_lim]
          [-Xsheet_count] [-Ssheet_size] [-Mmessage]
          [-Nnumber_up] [-Llayup_definition] [-Ksides]
          [-cdfghlmnpqrstvx] [filename...]
```

Explanation: You did not supply an argument with an option that requires one.

Action: Provide an argument to the option. Refer to the section in this chapter that discusses the specified option.

out of memory

Explanation: There is not enough memory to execute the **lpr** program.

Action: Check with the system administrator.

printer queue is disabled

Explanation: The system administrator disabled the printer queue with the **lpc** command.

Action: Select another printer queue or enter your command to this printer at a later time.

unknown printer 'printer'

Explanation: No entry for the printer exists in the **/etc/printcap** file.

Action: Check the spelling. Ask the system administrator to add an entry for the printer to the **/etc/printcap** file.

who are you?

Explanation: Your user ID is incorrect.

Action: Enter your correct user ID.

2.3 ULTRIX Print Queue Commands

Three ULTRIX commands allow you to manage print queues.

- **lpq** inspects print queues.
- **lprm** removes print jobs from a print queue.
- **lpc** is used for administration of print queues.

This section briefly describes each command and provides an example of using the command to manage a PrintServer queue. For further information, see **lpq**(1) and **lprm**(1) in the *ULTRIX Reference Pages* and **lpc**(8) in the *ULTRIX Programmer's Manual*.

2.3.1 lpq

The **lpq** command examines the spooling area used by the line printer daemon (**lpd**) for printing files on the printer and reports the status of jobs. The **lpq** command invoked without any arguments reports on any jobs currently in the default queue.

When jobs are being printed, **lpq** reports the queue as being "active." For each job submitted, **lpq** reports the user's name, current rank in the queue, the names of files comprising the job, the job identifier, and the total size in bytes. (The job identifier is a number that you can specify with the **lprm** command to remove a specific job.)

Example

This example submits a print job to a PrintServer queue called *script*. The **lpq** command is then used to inspect the spool queue for *script*.

```
% lpr -Pscript -Dascii .login
% lpq -Pscript
script is ready and printing via network
Rank      Owner    Job    Files      Total Size
active    rubin     80     .login     110 bytes
```

2.3.2 lprm

The **lprm** command removes one or more jobs from a printer's spool queue. Since the spooling directory is protected from users, using **lprm** is normally the only method by which you can remove a job. You need to specify the job identifier in the **lprm** command. (Use the **lpq** command to see what the job number is.)

Example

This example removes the job that was submitted to the script queue in the previous example by specifying the job number. The **lprm** command lists the print data file and control file that were deleted for print job 80.

```
% lprm -Pscript 80
dfA080rashi dequeued
cfA080rashi dequeued
```

2.3.3 lpc

A system administrator can use the **lpc** command to control the operation of the line printer system, including any PrintServer systems on the network. For each printer configured in **/etc/printcap**, an administrator or superuser can use **lpc** to perform the following functions:

- Start and stop printing and the spooler daemon for specified printers
- Enable and disable spooling of jobs in the spool queues
- Rearrange the order of jobs in the spool queue
- Report the status of printers and their associated spool queues and printer daemons

Example

This example submits a print job to a PrintServer queue called *script*. The **lpc status** command is then used to inspect the *script* spool queue.

```
% lpr -Pscript -Dascii filename
% su
.
.
.
# lpc status script
script:
    queuing is enabled
    printing is enabled
    1 entry in spool area
    daemon present
```

2.4 Setting Up POSTSCRIPT Print Jobs

You can control the content and format of a print job by creating and sending printer setup modules to a PrintServer system. Printer setup modules are usually written in the ANSI or POSTSCRIPT data syntax to perform print job setup functions, such as setting the page length for printing on legal-size paper (8.5 x 14 inches) or changing the default font for an ANSI print job.

You can store the printer setup modules that you create in any directory, and you can specify one or more setup modules to be copied to the PrintServer system before a file is printed.

With POSTSCRIPT setup modules, you can format the data on a page without modifying its content. Thus, you can reduce, rotate, or clip the page, but you cannot change its content. Using ANSI setup modules, you can modify the content of the page. For example, you can preserve an ANSI setup module environment for the next ANSI file in a job. Thus, you can change fonts, content, or page format.

You can send a POSTSCRIPT setup module to the PrintServer printer with a POSTSCRIPT print file. If either the print file or the setup module is not in POSTSCRIPT format, it must be translated into POSTSCRIPT before printing. Section 2.4.3 explains how to translate an ANSI file into POSTSCRIPT.

2.4.1 Creating Setup Modules

To create a printer setup module, follow these steps:

1. Using a text editor, create the setup module. (Some sample ANSI and POSTSCRIPT setup modules are shown in Section 2.4.2.)
2. Choose a directory in which to store the setup modules.

2.4.2 Sample Setup Modules

The following examples show setup modules that you can create and use to set up print jobs on a PrintServer system. (Refer to the *POSTSCRIPT Translators Reference Manual* for more information on using the ANSI data type.)

1. Sample ANSI Setup Module to Change Page Length

This setup module uses an ANSI escape sequence to set the page length for printing on legal-size paper (8.5 x 14 inches).

```
<ESC>[8 J
```

2. Sample ANSI Setup Module to Change Fonts

This setup module uses ANSI escape sequences to change the default font for an ANSI job from Courier to Elite italics.

```
<ESC>P1;14}relite0102sk00gg<ESC>\
<ESC>[3;14m
```

3. Sample POSTSCRIPT Setup Module to Print "Confidential"

This setup module uses a POSTSCRIPT file to print "Confidential" at the top of an 8.5 x 11-inch page.

```
% create confidential
/old-showpage /showpage load def % Copy the old definition.
/showpage {
  /Times-Roman findfont % this is the overlay
  14 scalefont
  setfont
  252 756 moveto
  (Confidential) show
  old-showpage
} bind def
showpage
```

2.4.3 Translating ANSI Files into POSTSCRIPT

PrintServer Client for ULTRIX software includes a shell script called **xlator_call**. The line printer daemon calls the **xlator_call** script to invoke the translator specified by the **data_type** option in the **lpr** command line or the **Da** entry in the **/etc/printcap** file. The line printer daemon passes parameters to the specified translator through the **xlator_call** script. (For more information on **xlator_call**(8), see the *ULTRIX Reference Pages*.)

You can use the **xlator_call** script to call the ANSI-to-POSTSCRIPT translator. Thus, you can translate your ANSI setup modules and print files into POSTSCRIPT.

Syntax

```
xlator_call datatype pagesize orientation width length indent
```

In this syntax, all parameters are required and must be entered in the order shown.

When the shell script is called, the path searched is:

```
/usr/local/lib/lpdfilters:/usr/ucb:/bin:/usr/lib:/usr/lib/lpfitlers
```

2.4.3.1 Examples

The following examples illustrate how to use POSTSCRIPT setup modules with POSTSCRIPT or ANSI print files and ANSI setup modules with ANSI print files.

1. This example uses the ULTRIX **cat** command to join a POSTSCRIPT setup module to the beginning of a POSTSCRIPT print file. The resulting single POSTSCRIPT file is then piped to the **lpr** command, as follows:

```
% cat post_setup.ps file.ps | lpr -Pprinter -Dpost
```

2. This example shows two steps in printing with a POSTSCRIPT setup module and an ANSI print file. First, it uses the **xlator_call** shell script to translate an ANSI print file into the POSTSCRIPT data syntax. The translated output is directed to a file called **ansi_file.ps**. Second, the POSTSCRIPT setup module is joined to the beginning of the translated ANSI print file. The resulting single POSTSCRIPT file is then piped to the **lpr** command, as follows:

```
% cat ansi_file | xlator_call ansi legal portrait 80 120 15 \  
> ansi_file.ps
```

```
% cat post_setup.ps ansi_file.ps | \  
lpr -Pprinter -Dpost
```

3. This example uses the **cat** command to join an ANSI setup module to the beginning of an ANSI print file. The resulting ANSI file is piped to the **xlator_call** shell script for translation to POSTSCRIPT. The translated output is then piped to the **lpr** command as follows:

```
% cat ansi_setup ansi_file \  
| xlator_call ansi legal portrait 80 120 15 \  
| lpr -Pprinter -Dpost
```

2.4.4 ARP Entry for Printing Over a TCP/IP Network

To avoid problems in printing from your PrintServer client to your PrintServer 20 over a TCP/IP network, as the PrintServer 20 does not respond to the ARP broadcast, add the following ARP entry in your **/etc/rc.local** file:

```
/etc/arp -s hostname ether_addr
```

The *hostname* is the PrintServer name, and *ether_addr* is the Ethernet address.

Your PrintServer administrator can obtain the Ethernet address by typing the following command on the TCP/IP PrintServer's supporting host and executing the configuration option:

```
# lprc hostname
```

The *hostname* is the PrintServer name.

2.4.5 Unknown Message from TCP/IP PrintServer

If you send a PostScript file containing a return status PostScript operator to a TCP/IP PrintServer, the following message is returned:

```
Message number %XNNNNNNNN
```

Because the `/usr/lib/lpfilters/lps_v3.a` device control archive uses a return status for reporting errors, you will receive an 8-digit hexadecimal message number instead of an error message from the TCP/IP PrintServer. For example, attempting to print duplex on a PrintServer 40 that does not support duplex printing returns message number `%X00000072`, instead of the message "PrintServer 40 does not support duplex printing."

Read the `/usr/examples/print/iplps_errors` file to find out the meaning of your message numbers and diagnose your problems.

2.4.6 ANSI Preamble Loading for TCP/IP PrintServers

The ANSI translator, **ansi_ps**, requires a preamble to be present in the printer, and is designed to send this to the printer. However, it is quicker if the TCP/IP supporting host software loads the preamble at boot time.

Follow these procedures to enable the TCP/IP supporting host software to load the preamble when it boots, and to disable the translator from sending the preamble:

1. Log in to your TCP/IP supporting host machine and enter superuser mode.
2. Move to the configuration directory for the TCP/IP supporting host software, copy the ANSI preamble file from an ULTRIX Version 4.2 system, and rename the file **lps_ansi_prologue.ps**:

```
# cd /usr/lib/iplps
# rcpnode:/usr/lib/lpfilters/preamble.ps lps_ansi_prologue.ps
```

The **node** is the name of the ULTRIX node.

3. Add the following line at the top of the **lps_ansi_prologue.ps** file:

```
/decbind {} def
```

4. Append the **lps_ansi_prologue.ps** file to the setup file for the printer.

The following command creates the file if it did not already exist:

```
# cat lps_ansi_prologue.ps >> setup.printername
```

5. Reboot your PrintServer.
6. Send all ANSI jobs to the PrintServer using the **fastansi** data type.

Using Layup Definition Files

This chapter explains how to create and use layup definition files. Chapter 2 shows how to include a layup definition file in a print job.

Layup is a PrintServer feature that can be associated with a PrintServer queue or used with the **lpr** command. To understand layup, it is important to remember the distinction between a sheet and a page. A sheet is a piece of paper. A page (sometimes called a logical page) is an image that you want to print, generally on a single sheet. For example, on this sheet, all the text from the chapter number to the page number is one page.

Using layup, you can control the mapping of pages to sheets so that you can specify where on a sheet a page is printed. Layup also lets you set the number of pages printed on a sheet. You can use layup with any file that can be printed on a PrintServer system. Layup is implemented through the `number_up` and `layup_definition` options to the **lpr** command. (See Chapter 2, *Printing on a PrintServer System*, for more information on using these options.)

The `layup_definition` option calls a layup definition file. You can use layup definition files to set sheet margins, page borders, a page grid, as well as a number of other options. You can associate a layup definition file with a PrintServer queue by editing the `/etc/printcap` file. Chapter 3, *Configuring the Client*, in the *Installation Guide: PrintServer Client for ULTRIX* explains how to make this entry in the `/etc/printcap` file.

3.1 Creating a Layup Definition File

By convention, layup definition files are stored in the directory `/usr/lib/lpdfilters` and end with characters `.lup`.

When creating a layup definition file, follow these rules:

- Give the layup definition file a valid ULTRIX file name.
- Start each option on a new line. Blank lines and white space within lines are ignored.
- Lines beginning with an exclamation point (!) are considered comments and are ignored.
- Do not abbreviate options.
- Options are not case sensitive, so you can type them either in uppercase or lowercase.
- Some options take values, indicated by an equal sign (=). You must use at least one value after an equal sign. Separate multiple values with commas.
- If you specify an option more than once in a layup definition file, the last setting is the one that layup uses.

Table 3–1 lists the options you can use in a layup definition file. The following sections describe the layup options that are set in a layup definition file and show some sample layup definition files.

Table 3–1 Layup Definition File Options

Borders	Draws a border around each page.
Margins	Sets the margins for a sheet. Pages are scaled to fit within margins.
Alternate	Exchanges a pair of margins on alternate sheets.
PagesPerSheet	Sets the number of pages that will print per sheet, no matter how many page spots are set by the <code>number_up</code> option.
FirstPage	Specifies the first page spot where a page will be printed.
PageOrder	Specifies the order in which pages appear on a sheet.
Grid	Sets the number of page spots on a sheet in columns and rows.

3.1.1 Borders

The **Borders** option draws a border around each page. Borders are useful for differentiating pages when you are printing a job using the `number_up` option. Borders are drawn only around actual pages, not around all page spots. This allows you to distinguish between blank pages in a document and nonexistent pages. Figure 3–1 shows pages printed using the **Borders** option. Figure 3–2 shows the same pages printed without the **Borders** option.

You can turn off the borders, using the option **NoBorders**. If you use the `number_up` option, the default is to draw borders.

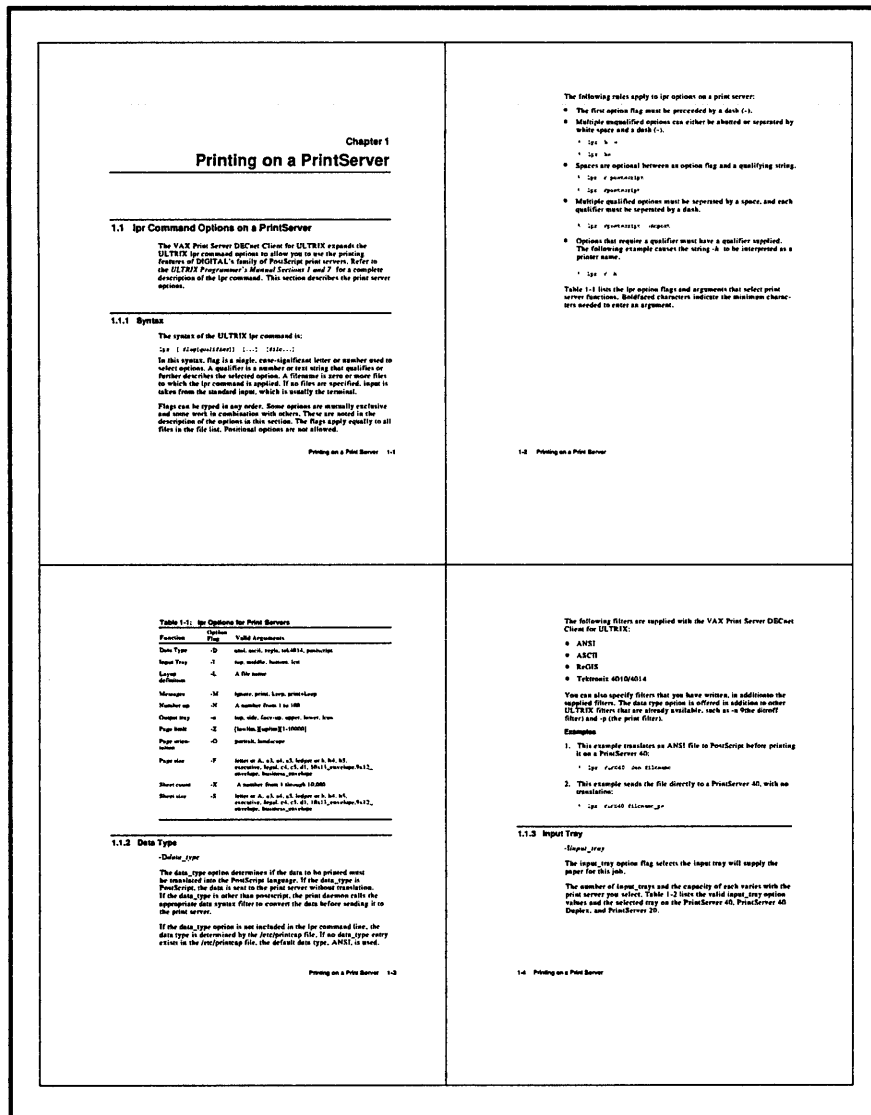
3.1.2 Sheet Margins

The **Margins** option sets the margins of the sheet on which you are printing. This option takes four values, which set the top, bottom, left, and right margins for a sheet. (See Figure 3–3.) The numbers, or margin values, are given in printer's points. (There are 72 points to an inch.)

You always specify the values for sheet margins in the order top, bottom, left, and right, because the sheet margins are independent of whether the page is portrait or landscape. Positive values start at the edge of the page and move toward the center; negative values start at the edge of the page and move away from the center.

When you print using a layup definition file that uses the **Margins** option, the page is scaled to fit on the area of the sheet that remains after margins are subtracted. This means that the size of the text and graphics you are printing may be reduced. You can use negative margin values to move the page on the sheet without scaling down the size of the page. (See the sample layup definition file in item 5. in Section 3.2.)

Figure 3-1 Four Pages to a Sheet (4-Up) with Borders



MLO-004318

If you set the top and bottom, or right and left, margins so that they touch, jobs that contain the layout file will not print. If you set the top and bottom, or right and left, margins so that they overlap, you must also set a page grid, or jobs that contain the layout file will not print.

If you specify the **Margins** option without specifying any values, the translator uses a value of 36 for all four margins. If you specify **NoMargins**, the translator uses the value 0 for all four margins.

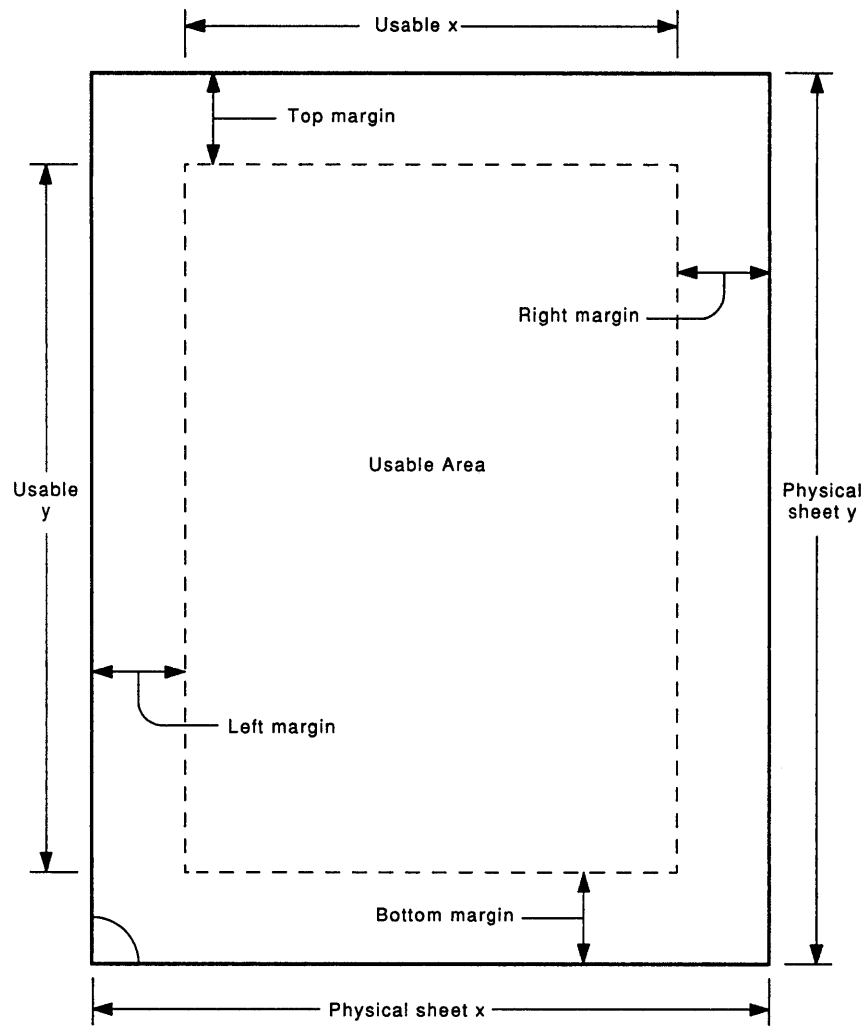
The default margin setting is a 36-point (a half-inch) margin on all sides.

[illegible]

Example

Margins = 10, 10, 60, 10

Figure 3–3 Usable Sheet Area with Margins Option



MLO-001474

3.1.3 Margin Alternation

The **Alternate** option lets you exchange a pair of margins on alternate sheets. This option is useful for preparing a job for double-sided copying or printing.

The following list describes the values that can be supplied to the **Alternate** option.

Value	Result
LEFT	Left and right margins alternate
RIGHT	Left and right margins alternate
None	Left and right margins alternate
TOP	Top and bottom margins alternate
BOTTOM	Top and bottom margins alternate

If you specify **NoAlternate**, neither pair of margins alternates.

If you do not specify this option, neither pair of margins alternates, unless you print using the sides duplex option. If your job allows you to select format at the time you print, and you specify duplex for the sides option, the margins alternate. If you specify duplex for the sides option and include a layup definition file that specifies **NoAlternate**, the margins do not alternate.

3.1.4 Pages per Sheet

The `number_up` option to the **lpr** command determines the maximum number of pages that you can print for each sheet. The **PagesPerSheet** option in a layup definition file specifies how many pages are actually printed on a sheet.

Because `number_up` determines the number of page spots, it also determines the layout of the page spots on the sheet. **PagesPerSheet** determines only the range of page spots to use, not their layout.

You must supply a value to **PagesPerSheet**, and the value must be less than the value supplied to `number_up`. You cannot specify **NoPagesPerSheet**.

The default is to use all the page spots on a sheet.

Example

Assume you have a layup definition file called **sixpages.lup** that contains the following line:

```
PagesPerSheet = 6
```

You print a file using the following command:

```
% lpr -L /usr/lib/lpddfilters/sixpages.lup -N8 filename
```

Even though this command provides eight page spots for each sheet, the job prints with six pages on a sheet.

3.1.5 First Page

The **FirstPage** option specifies the first page spot on a sheet where a page will be printed.

You must specify a value to **FirstPage**. The value is the number of the page spot on which to print the first page. Page spots are numbered starting at 1. The value supplied to **FirstPage** must be less than the value supplied to `number_up`. You cannot specify **NoFirstPage**.

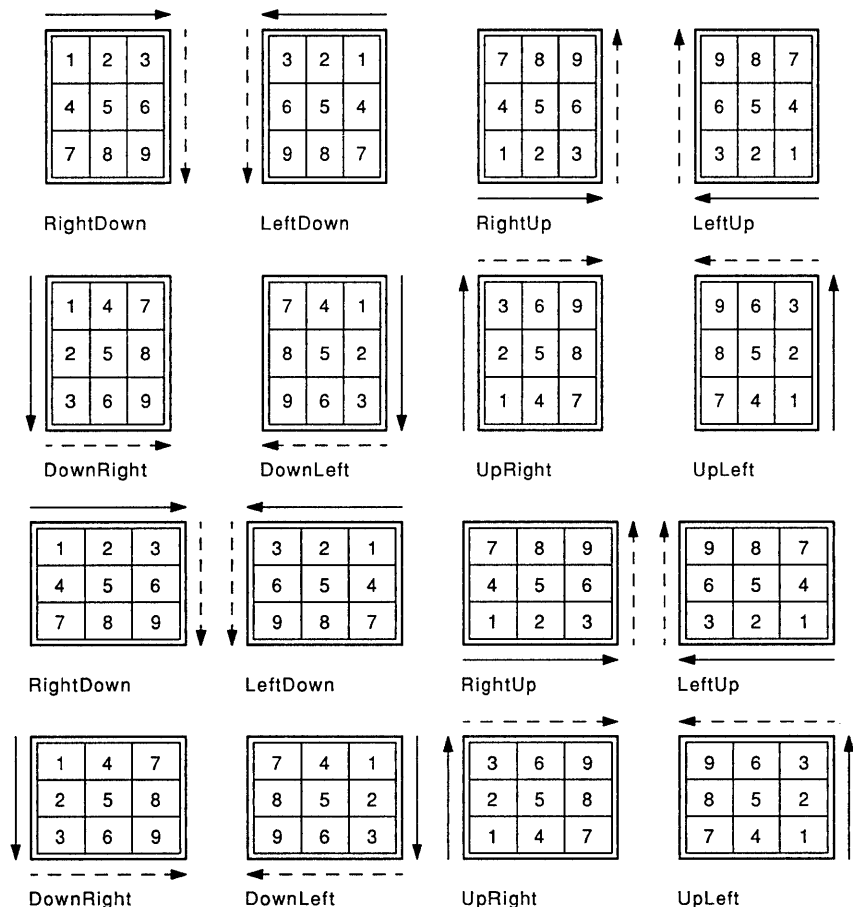
The default is to use the first page spot on a sheet.

The **FirstPage** option affects only the first sheet of the print job. All other sheets begin printing at page spot 1.

3.1.6 Page Order

The **PageOrder** option sets the order in which the pages appear on the sheet. This can be thought of as the reading order of the pages. (See Figure 3–4 for examples of page order.)

Figure 3–4 Examples of Page Order Option (-N9)



MLO-001475

You must supply a value to **PageOrder**. Table 3–2 describes the eight acceptable values. If you do not use the **PageOrder** option, the page order is RightDown.

Table 3–2 Values for the PageOrder Option

Value	Description
RightDown	Pages are numbered left to right, top row first, then moving down
LeftDown	Pages are numbered right to left, top row first, then moving down

(continued on next page)

Table 3–2 (Cont.) Values for the PageOrder Option

Value	Description
RightUp	Pages are numbered left to right, bottom row first, then moving up
LeftUp	Pages are numbered right to left, bottom row first, then moving up
DownRight	Pages are numbered top to bottom, left column first, then moving right
DownLeft	Pages are numbered top to bottom, right column first, then moving left
UpRight	Pages are numbered bottom to top, left column first, then moving right
UpLeft	Pages are numbered bottom to top, right column first, then moving left

3.1.7 Page Grid

Instead of using the `number_up` option to specify the maximum number of page spots on a sheet, you can use the **Grid** option. The **Grid** option lets you specify the number of page spots on a sheet in columns and rows.

Grid takes two values. The first value is the number of columns in the grid; the second is the number of rows. The product of the two values cannot exceed 100.

The grid is interpreted in relation to the pages, not the sheet. As with `number_up`, the orientation of the pages is independent of the sheet orientation. This means that you can print portrait-shaped pages, but you hold the sheet in landscape position to read them.

Example

This example specifies that the pages will print 6-up, in a grid of two columns and three rows.

```
Grid = 2, 3
```

3.2 Sample Layup Definition Files

The first four sample layup definition files in this section are included as part of the client software. They are located in the directory `/usr/lib/lpddfilters`.

1. The following sample layup definition file is for a print job that will be printed one page to a sheet. It sets a wide left margin, so the sheets can be punched. It does not alternate the margins, and it turns off borders. Because it sets margins that are smaller than the sheet margins, text may be scaled to fit the available area.

```
! lpssingleholes.lup specifies a larger left margin to allow for
! hole punching on single-sided sheets. It is meant for printing
! one page per sheet.
```

```
no borders
margins = 19, 19, 60, 19
no alternate
```

2. The following sample layup definition file is for a print job that will be copied double-sided and then punched. It alternates the larger margin. It turns off the borders, as it is intended for jobs that are printed one page to a sheet. Because it sets margins that are smaller than the sheet margins, text may be scaled to fit the available area.

```
! lpsdoubleholes.lup specifies a larger left margin to allow for
! hole punching. It alternates the larger margin for double-sided
! copying. It is meant for printing one page per sheet.
```

```
no borders
margins = 19, 19, 60, 19
alternate = left
```

3. The following sample layup definition file is for jobs that are printed with more than one page to a sheet, for example, a job printed with `number_up=6`. This file uses borders to differentiate among pages. Again, it leaves a wide left margin but does not alternate margins. Because it sets margins that are smaller than the sheet margins, text may be scaled to fit the available area.

```
! lpsnup.lup can be used with any number specified for number up.
! It specifies page borders and allows for hole punching. It does
! not alternate margins, because it is not meant for double-sided
! copying.
```

```
borders
margins = 19, 19, 60, 19
```

4. The following sample layup definition file is for jobs that print one page to a sheet. It sets a wide left margin, so sheets can be punched. If you select two-sided (duplex) formatting, it alternates the larger margin. If you select one-sided (simplex) formatting, margins do not alternate.

```
! lpsholes.lup provides for a larger left margin for hole punching
! that will work on either duplex or simplex modes. Note that it
! will not produce the desired result with any tumbling operation.
```

```
no borders
margins = 19, 19, 60, 19
```

5. The following sample layup definition file is for jobs that print one page per sheet. It allows a wide margin for hole punching. It maintains the full size of the page image by using a negative right margin. The negative right margin moves some of the page image off the physical sheet, so this layup definition file is good only for pages that have page margins.

```
! lpsshiftforholes.lup specifies a wide left margin for hole
! punching without scaling down the size of the page image.
! The negative margin means some of the page image is off the
! physical sheet. It is meant for printing one page per sheet.
```

```
no borders
margins = 0, 0, 60, -60
```

Device Control Library

The device control library is an ULTRIX archive file that contains modules of POSTSCRIPT procedures. These modules implement the **lpr** command line options that affect output appearance. This chapter describes the device control library and explains how to include device control modules in a print job.

The device control library that is part of the PrintServer Client software is stored in a single archive file named **/usr/lib/lpddfilters/lps40.a**. The PrintServer client software includes two device control libraries. You must use the device control library that is compatible with the version of PrintServer software that is running on your supporting host:

- The **/usr/lib/lpddfilters/lps40.a** archive contains the device control modules for PrintServer Versions 2.0 and 2.1.
- The **/usr/lib/lpddfilters/lps_v3.a** archive contains the device control modules for PrintServer Version 3.0.

You can use the ULTRIX archive (**ar**) utility to extract a device control module from the archive file for editing with any text editor available on your system. You can then replace the module in the archive. For further information, see **ar(1)** in the *ULTRIX Reference Pages*.

The device control library name is specified in the **DI** entry in the **/etc/printcap** file. Chapter 3, *Configuring the Client*, in the *Installation Guide: PrintServer Client for ULTRIX* lists the default values for PrintServer systems in the **/etc/printcap** file.

The names of modules supplied by Digital that are stored in the device control library begin with the characters **LPS_**.

The line printer daemon extracts and sends some device control modules to the PrintServer system, outside the control of the user or the system administrator. For example, the line printer daemon sends device control modules at the start of each print job to set up the printer.

A system administrator who sets up a print queue can associate printer capabilities with that queue. A user can also associate printer capabilities with a print queue by specifying options to the **lpr** command. The line printer daemon loads one or more device control library modules as required or supplies different arguments to device control library modules that are automatically loaded.

4.1 Device Control Modules

One of the device control modules that the line printer daemon sends to the PrintServer system is a POSTSCRIPT procedure that generates a print job's flag page. Other modules implement the following options to the **lpr** command:

input_tray
number_up

orientation
output_tray
page_limit
page_size
sheet_count
sheet_size
sides

These device control modules reside in the printer until the print job terminates. They modify the behavior of a print job in two ways:

1. By using POSTSCRIPT extension operators.

The device control modules use the POSTSCRIPT extension operators to control the operation of the printer. These operators are unique to a specific printer model or a specific class of printers.

2. By redefining native POSTSCRIPT operators.

The device control modules may redefine native POSTSCRIPT operators to modify a print job's behavior. This programming technique is called *layering*; it is made possible by the way POSTSCRIPT performs implicit name searches. See *Programmer's Supplement: POSTSCRIPT Printers* for more information about layering.

The following list describes the POSTSCRIPT modules provided in the device control library for the client software. You should not remove, rename, or modify these modules.

LPS_DECMCSENCODING	The encoding vector for DEC MCS (Digital Multinational Character Set).
LPS_ERRORHANDLER	Procedure that contains strings used for POSTSCRIPT error reporting. LPS_ERRORHANDLER prints any partial bitmaps not imaged by a showpage operator.
LPS_EOJ	Used as a reset module.
LPS_FINDFONT_ISOLATIN1_DECMCS_V40	Procedure that extends the findfont operator to recognize ISO Latin-1 and DEC MCS character encodings.
LPS_ISOLATIN1ENCODING	The encoding vector for the ISO Latin-1 Character Set. This is meant for use on POSTSCRIPT printers that use a version of POSTSCRIPT earlier than Version 40.
LPS_FLAGPAGE	Code for the flag page.
LPS_FLUSHPAGES	Used to flush the last pages of a job when layup is used.
LPS_LOADDICT	Used to set the initial POSTSCRIPT state for some of the other modules. It creates and initializes a dictionary shared by the LPS_ modules.
LPS_SETCONTEXT	Used to establish the proper context for a user's job.
LPS_SETINPUTTRAY	Selects the input tray to be used for a job.
LPS_SETNUMBERUP	Sets the number of pages to be printed on a sheet.
LPS_SETOUTPUTTRAY	Selects the output tray to be used for a job.
LPS_SETPAGELIMIT	Sets the range of pages to be imaged for a job.
LPS_SETPAGEORIENTATION	Sets the page orientation for a job.
LPS_SETPAGESIZE	Sets the page size for a job.

LPS_SETSHEETCOUNT	Sets the number of times each sheet is to be reprinted for a job.
LPS_SETSHEETSIZE	Establishes the correspondence between the user's data and the sheet size.
LPS_SETSIDES	Sets the side of the paper for duplex printing.

4.2 Order of Device Control Modules in a Print Job

Table 4–1 lists each device control module that can be included in a print job, whether it is optional or required, and the source of the request to send that data.

Table 4–1 Order of Data Sent to a Print Job

Data Sent to Printer	Status	Origin of Request
LPS_SETOUTPUTTRAY	Optional	lpr -o or Ot in /etc/printcap
LPS_FLAGPAGE	Optional	Loaded by lpd except with lpr -h or sh in /etc/printcap
LPS_LOADDICT	Required	lpd
LPS_SETNUMBERUP	Optional	lpr -N or Nu in /etc/printcap
LPS_SETSIDES	Optional	lpr -K or Si in /etc/printcap
LPS_SETINPUTTRAY	Optional	lpr -I or It in /etc/printcap
LPS_SETPAGEORIENTATION	Optional	lpr -O or Or in /etc/printcap
LPS_SETPAGESIZE	Optional	lpr -F or Ps in /etc/printcap
LPS_SETSHEETSIZE	Optional	lpr -S or Ss in /etc/printcap
LPS_SETSHEETCOUNT	Optional	lpr -X
LPS_SETPAGELIMIT	Optional	lpr -Z or Ul in /etc/printcap
Output from layup translator	Optional	lpr -L or Lf in /etc/printcap
LPS_SETCONTEXT	Required	lpd
User's first print file		lpr
User's second print file		lpr
.		
.		
.		
LPS_FLUSH_PAGES	Optional	lpr -N or Nu in /etc/printcap
LPS_EOJ	Required	lpd

4.3 Creating a Device Control Module

You can add modules to the device control library. Modules sent directly to the PrintServer network printer bypassing the translators must contain POSTSCRIPT code. When creating a new POSTSCRIPT device control module, follow the procedure in this section. If you are creating ANSI device control modules, follow the procedure in Section 2.4.

1. Design the POSTSCRIPT code for the desired function.
2. Test the new function in a controlled environment.

3. Add the code to the appropriate archive (**/usr/lib/lpddfilters/lps_v3.a** for supporting hosts running PrintServer software, Version 3.0; **/usr/lib/lpddfilters/lps40.a** for supporting hosts running PrintServer software, Version 2.0 or 2.1).
4. Edit the **/etc/printcap** file to associate the new archive file with a print queue.
5. If the queue is busy, stop it with the **/etc/lpc** command. Then, restart the queue using **/etc/lpc**. Do this for every queue that uses this library.

This chapter describes how to handle special types of media on PrintServer network printers.

5.1 Letterhead Paper

This section describes how to load letterhead paper in PrintServer network printers.

5.1.1 PrintServer 20 Network Printer

When a PrintServer 20 printer prints on both sides of a sheet, it prints the second side first, then flips the sheet over and prints the first side. Because of this, you load letterhead paper differently depending on whether you are printing on one or both sides of a sheet.

- If you are printing on one side of a sheet, load the paper with the letterhead side face up and the letterhead to the right.
- If you are printing on both sides of a sheet, load the paper with the letterhead side face down and the letterhead to the right.

5.1.2 PrintServer 40 Series Network Printers

The PrintServer 40 and the PrintServer 40 Plus printers support only one-sided printing. Always load letterhead paper with the letterhead side face up and the letterhead to the right.

5.2 Drilled Paper

This section describes how to load drilled (hole-punched) paper in PrintServer network printers.

5.2.1 PrintServer 20 Network Printer

When a PrintServer 20 printer prints on both sides of a sheet, it prints the second side first, then flips the sheet over and prints the first side. Because of this, you load drilled paper differently depending on whether you are printing on one or both sides of a sheet.

- If you are printing on one side of a sheet, load the paper with the drilled edge as the leading edge, that is, the edge closer to the printer.
- If you are printing on both sides of a sheet, load the paper with the drilled edge away from the printer.

5.2.2 PrintServer 40 Series Network Printers

The PrintServer 40 and the PrintServer 40 Plus printers support only one-sided printing. Always load drilled paper with the drilled edge as the leading edge, that is closer to the printer.

5.3 Transparencies

This section describes how to print transparencies on PrintServer network printers.

5.3.1 PrintServer 20 and PrintServer 40 Series Network Printers

When you print transparencies on the PrintServer 20, PrintServer 40, or PrintServer 40 Plus model, send the output to the side output tray, stacked face-up. This allows the transparencies to travel through the printer on a straight path.

If you set up a special queue for printing transparencies, you can associate the `OUTPUT_TRAY=FACE_UP` parameter with the queue.

5.4 Gummed Labels

This section describes how to print gummed labels on PrintServer network printers.

5.4.1 PrintServer 20 and PrintServer 40 Series Network Printers

When you print on label stock on a PrintServer 20, a PrintServer 40, or a PrintServer 40 Plus model, send the output to the side output tray. This allows the labels to travel through the printer on a straight path.

If you set up a special queue for printing labels, you can associate the `OUTPUT_TRAY=FACE_UP` parameter with the queue.

5.5 Preprinted Forms

You cannot make photocopies of a form and use the copies as paper on PrintServer network printers. The toner on the photocopied forms melts and damages the fuser in the printer.

Error Logging and Accounting

This chapter describes error messages and accounting on a PrintServer system. The client communications filters extract and return user and operator error messages as well as accounting information about each print job from PrintServer software to the line printer daemon.

6.1 Error Logging

User messages are sent in response to either your print request or a system condition that affects your print request. User messages are ignored unless you specify the **-Mkeep** option to the **lpr** command. When you specify **-Mkeep**, **lpd** mails the user messages to you.

Operator messages are sent to **/dev/console**, unless you specify an error log file for the **lf** entry in the **/etc/printcap** file. These messages are then directed to the specified log files.

6.1.1 Message Format

PrintServer messages have the following format:

```
lpscomm: <printer node name>: message text
iplpscomm: <printer node name>: message text
```

lpscomm is the PrintServer Client for ULTRIX filter that distributes the error messages from the PrintServer system to the line printer daemon. **iplpscomm** handles this function on a TCP/IP network. The *printer node name* is the PrintServer node from which the error message was generated. *Message text* is the message string.

6.1.2 Message Descriptions

The message descriptions in this section are alphabetized by the first word in the message text.

Bad password.

Explanation: An invalid password has been sent to the printer.

User Action: Change the password to the one used by the printer, and resubmit the job.

condition on line *line number* in layup definition.

Explanation: There is an error in the layup definition file you have included in your print job.

User Action: Check the line in the layup definition file that is indicated in the error message. Section 6.1.2.1 describes the error conditions that can appear in a layup definition file.

dictfull: No more room in dictionary—offending command is *string*

Explanation: The POSTSCRIPT interpreter detected this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

dictstackoverflow: Too many begins

Explanation: The POSTSCRIPT interpreter has sensed too many *begins* without corresponding *ends*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

dictstackunderflow: Too many ends

Explanation: The POSTSCRIPT interpreter has sensed too many *ends* without corresponding *begins*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

Duplex to "face_up" output tray is not supported on *PrintServer-name*

Explanation: You requested two-sided printing to be sent to the face-up output tray, which is an unsupported combination.

User Action: Print the job again, using either one-sided printing and the face-up output tray or using two-sided printing and a different output tray.

execstkov: Exec nesting is too deep—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

exitserver has been executed—permanent state may be changed

Explanation: The POSTSCRIPT operator **exitserver** has been successfully executed. The permanent POSTSCRIPT system parameters may have been altered. This message is also displayed during the first job after server startup or after a server manager has changed a job default.

User Action: If this message appears at any time except for the above conditions, notify the server manager.

interrupt: The job has been interrupted

Explanation: The POSTSCRIPT interpreter has sensed an external request to interrupt the POSTSCRIPT program. This message also appears as part of the Abort Job process.

User Action: No further action. This message confirms a requested action.

invalidaccess: Attempt to store into readonly object—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

invalidexit: Exit not in loop

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

invalidfileaccess: Bad file access string—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

invalidfont: Bad font name or dictionary—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

invalidrestore: Improper restore—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

ioerror: System I/O error occurred—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

Job aborted due to PrintServer protocol error

Explanation: The client or the PrintServer printer violated the *Local Area Print Station (LAPS) protocol*, causing the job to be aborted.

User Action: Try submitting the job again. If necessary, reboot the PrintServer system. If the job still fails, submit a Software Performance Report (SPR) with a description of your PrintServer configuration and the circumstances under which this error occurred.

Job *jobnumber* aborted

Explanation: The specified job was aborted for one of the following reasons:

1. You deleted an active job on the local PrintServer queue.
2. The POSTSCRIPT interpreter requested that the job be aborted due to a printer or internal error.
3. The logical link between the printer and the client filter was broken.
4. The client or **lpscomm** process crashed.
5. An internal error occurred in the LAPS job controller.

User Action: Take the following corresponding action:

1. None.

2. Another error message accompanies the abort job message. That message gives you specific information about the required action.
3. Check that the DECnet network is functioning properly between the PrintServer system and the client system.
4. Submit a Software Performance Report (SPR) with a description of your PrintServer configuration and the circumstances under which this error occurred.
5. Take the same action as in 4. above.

Job *jobnumber* aborted by Server Management

Explanation: The specified job on the PrintServer queue was aborted by the server management Abort Job command.

User Action: None.

Job *job number* Connect

Explanation: The PrintServer system has accepted job *job number* for processing.

User Action: None.

Layup_definition parameter ignored, since Number_up=0 was specified

Explanation: Specifying a number_up value of 0 (-N0) disables layup.

User Action: If you want to use a layup definition file, either specify a value for number_up or do not use the number_up option. The default is number_up=1.

limitcheck: Implementation limit exceeded—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

nocurrentpoint: Path is empty—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

Normal, successful completion

Explanation: This message indicates that the print job completed successfully.

User Action: None.

One_sided_duplex is not supported on *PrintServer-name*

Explanation: Your PrintServer system does not support the value *one_sided_duplex* for the sides option (-K).

User Action: Reissue the **lpr** command without this option.

Output tray *output-tray-name* not supported for *printer-name*.

Explanation: The PrintServer printer you are using does not have the output tray you requested.

User Action: Reissue the **lpr** command and request a different output tray. Table 2–7 in Chapter 2 lists the supported output trays.

Pending job *jobnumber* aborted

Explanation: A job that was pending on the PrintServer queue has been removed from the queue by the server management **Abort Job** command.

User Action: None; this confirms the requested action.

PrintServer is currently disabled and cannot accept jobs

Explanation: You submitted a print job for a PrintServer system that is currently not accepting jobs.

User Action: Ask your PrintServer system manager to enable job acceptance on that PrintServer system, using the server management console.

Problem Unknown

Explanation: The POSTSCRIPT interpreter has encountered an irrecoverable error.

User Action: Reboot the PrintServer system and submit a Software Performance Report (SPR).

Please reboot the PrintServer - FATAL ERROR

Explanation: The POSTSCRIPT interpreter has encountered an irrecoverable error.

User Action: Reboot the PrintServer system and submit a Software Performance Report (SPR).

Recoverable network error—print job *jobnumber* queued

Explanation: A network error has occurred, which caused your print job to stop printing. Your job will be queued.

User Action: Check the output to make sure that all the files in the job were printed. Your job will be reprinted from the beginning.

rangecheck: Argument out of bounds—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

Rest of job (to EOJ) will be ignored

Explanation: An error or some other event has occurred that is causing POSTSCRIPT to ignore the rest of the job. One or more messages preceding this one should provide information on why the job is being flushed.

User Action: Check the other messages that accompanied this one.

Side tray is closed on *PrintServer name*

Explanation: You have requested that your output be sent to the side output tray; however, the tray is closed.

User Action: Open the side output tray.

stackoverflow: Operand stack overflow—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

stackunderflow: Operand stack underflow—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

String is too long—it has been truncated

Explanation: A string passed internally is too large for the receiving buffer.

User Action: This is a software problem; submit a Software Performance Report (SPR).

syntaxerror: Input ended in string or procedure body—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

timeout: Time limit exceeded

Explanation: A POSTSCRIPT program has executed beyond the time limit set by the server manager. This condition can be caused by either an unusually complex page definition or an error in the POSTSCRIPT program that results in an infinite loop.

User Action: If the message is a result of a complex page definition, ask the server manager to increase the value of the job time-out limit.

If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

Timeout trying to send a message

Explanation: The timeout period expired before server management could send a message to the printer.

User Action: Try the operation again.

Tray selection is not supported by multi-page layup.

Explanation: Tray selection operators are ignored when you use layup to print more than one page to a sheet.

User Action: None required. If you require output from the tray you selected, reissue the **lpr** command without specifying multipage layup.

typecheck: Argument of wrong type—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by DIGITAL, submit a Software Performance Report (SPR).

undefined: Name not known—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed the error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

undefinedfilename: File not found—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by DIGITAL, submit a Software Performance Report (SPR).

undefinedresult: Number overflow or underflow—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

unmatchedmark: Expected mark not on stack—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

unregistered: POSTSCRIPT has encountered a system error—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

Unrecoverable network error on job *jobnumber*—queue restart required

Explanation: A network error has occurred, which has caused the print queue to stop printing.

User Action: The system manager should restart the queue.

USERDATA, *string*

Explanation: The POSTSCRIPT program requested that this data be printed (POSTSCRIPT operators **print**, **pstack**, **=**, **==**).

This message is also used to notify a user that some part of the print request cannot be executed. For example, you see this message if you specify an executive sheet size when there is no executive size paper in any of the paper trays.

User Action: If the text of the message refers to an option in the **lpr** command, change the command line to comply with the PrintServer system or change the PrintServer system to comply with the **lpr** command. Then resubmit the print request.

VMerror: POSTSCRIPT virtual memory exhausted—offending command is *string*

Explanation: The POSTSCRIPT interpreter sensed this error while trying to execute the POSTSCRIPT command represented by *string*.

User Action: If this error occurred during the printing of a file with a data type other than POSTSCRIPT, a filter has an error.

If the error occurred during the printing of a POSTSCRIPT file, either the POSTSCRIPT file or the application that generated the POSTSCRIPT file is in error. If the application is supplied by Digital, submit a Software Performance Report (SPR).

6.1.2.1 BADLAYDEF Messages

This section describes some of the messages you receive with the message identification BADLAYDEF. The PrintServer system identifies POSTSCRIPT errors, including **lpr** options and layup definition file errors, as BADLAYDEF messages.

For layup definition files, messages appear in the following format:

lpscomm: <printer node name>: BADLAYDEF: <condition> on line
<line-number> in layup definition

The *condition* describes the problem, and *line-number* is the line number in the layup definition file on which the error occurred.

Bad form for margin values

Explanation: The values given for the **Margins** option are not valid.

User Action: Refer to Chapter 3 for information on the **Margins** option.

Bad form for page grid values

Explanation: The values given for the **Grid** option are not valid.

User Action: Refer to Chapter 3 for information on acceptable values for the **Grid** option.

Cannot give values with negated option 'option'

Explanation: A line contains both a negated option and a value; for example, **NoAlternate=left**.

User Action: Edit the layup definition file and reissue the **lpr** command.

Cannot negate option 'option'.

Explanation: An option is negated that cannot be negated.

User Action: Edit the layup definition file and reissue the **lpr** command.

Cannot supply a value for 'option'

Explanation: A value is given for an option that cannot take a value.

User Action: Edit the layup definition file and reissue the **lpr** command.

"coppage" is not supported by multi-page layup.

Explanation: The **coppage** operator is ignored when you use layup to print more than one page to a sheet.

User Action: None.

Could not find a number as a value

Explanation: The value given for an option must be a number.

User Action: Edit the layup definition file and reissue the **lpr** command.

error on line *line number* in layup definition.

Explanation: The layup definition file that you have included in your print job has an error.

User Action: Check the line in the layup definition file that is indicated in the error message. Chapter 3 of this manual describes the layup definition options.

Extra characters present after values

Explanation: Legitimate values for an option are followed by extra characters.

User Action: Edit the layup definition file and reissue the **lpr** command.

Fatal device control library problem. Config error: *configuration error* on *PrintServer name*.

Explanation: This fatal error indicates a serious problem with the device control library.

User Action: This error should not occur. Submit a Software Performance Report (SPR).

First page is greater than pages per sheet. First page set to 1.

Explanation: You have included a layup definition file in your print job that uses the **FirstPage** and **PagesPerSheet** options. However, the **FirstPage** option requests that the first page be printed in a page spot beyond the number specified with the **PagesPerSheet** option.

User Action: No action is necessary if you want the first page to be printed in the page spot given with the **PagesPerSheet** layup option. For example, if you specify **PagesPerSheet=5**, the first page on the first sheet will print in the fifth page spot. If you want the first page to print at a different spot, however, edit the layup definition file and reissue the **lpr** command.

First page is greater than pages per sheet. First page set to pages per sheet.

Explanation: The number specified with the **FirstPage** option in the layup definition file is greater than the number specified with the **number_up** option. The first page defaults to the number specified with **number_up**.

User Action: No action is necessary if you want the first page set to the number specified with the **number_up** option. Otherwise, edit the layup definition file and reissue the **lpr** command.

Input tray selection not supported for the *printer name*.

Explanation: You have requested an input tray that does not exist on this printer.

User Action: Reissue the **lpr** command, without the request for an input tray.

Layup definition margins overlap one another.

Explanation: You have included a layup definition file in your print job that sets margins that overlap.

User Action: See Chapter 3 for information on the **SheetMargins** option in layup definition files.

Layup definition margins result in no usable sheet area.

Explanation: You have included a layup definition file in your print job that sets the margins in a way that leaves no area to print.

User Action: See Chapter 3 for information on the **SheetMargins** option in layup definition files.

media size size medium is not in the *input tray name* tray in *PrintServer-name*.

Explanation: Your PrintServer system does not have the paper you requested in the input tray you requested.

User Action: Reissue the **lpr** command, specifying only the paper size or the input tray. Or load the paper you requested in the input tray you requested and reissue the **lpr** command.

media size size medium is not in the *PrintServer name*

Explanation: Your PrintServer system does not have the paper you requested in the input tray you requested.

User Action: Reissue the **lpr** command, specifying only the paper size or the input tray. Or, load the paper you requested in the input tray you requested and reissue the **lpr** command.

Must express a value for option 'option'

Explanation: No value is given for an option that requires a value.

User Action: Edit the layup definition file to include the required value and reissue the **lpr** command.

No option present

Explanation: A line is not blank, but it also does not contain an option.

User Action: Edit the line in the layup definition file and reissue the **lpr** command.

No values specified for option 'option'

Explanation: A line contains an equal sign (=) but no values.

User Action: Edit the layup definition file to include the value or remove the equal sign. Reissue the **lpr** command.

Number must be greater than 1

Explanation: The numeric values for an option are out of range, for example **PagesPerSheet=0**.

User Action: Edit the layup definition file to include the correct value and reissue the **lpr** command.

Output will be delivered to the *output tray* tray on *PrintServer name*.

Explanation: Not all PrintServer systems have the same output trays. If you request a supported output tray that is not available on your PrintServer system, the output is sent to a different tray.

User Action: None. This message is informational.

Page count must be less than 100

Explanation: The **Grid** option must have two positive values. The product of the values must be 100 or less.

User Action: Edit the layup definition file so that the product of the values for the **Grid** option is 100 or less and reissue the **lpr** command.

Pages per sheet > Number_Up. Continuing with pages per sheet = Number_Up.

Explanation: You have included both a layup definition file and the number_up option in your print job. However, the layup definition file requests a number of pages printed per sheet that is greater than the number specified with the number_up option.

User Action: No action is necessary if you want the number of pages per sheet to equal the number you specified with the number_up option. If you want the number of pages per sheet to equal the number given in the layup definition file, reissue the **lpr** command, specifying a larger number with the number_up option.

PrintServer name does not support duplex printing.

Explanation: The PrintServer system you are using does not support duplex printing. If you request both tumble and duplex printing and neither is supported, you see only the error message stating that tumble printing is not supported.

User Action: Send your print job to a PrintServer system that supports duplex printing, or reissue the **lpr** command without requesting duplex printing.

PrintServer name does not support tumble printing.

Explanation: The PrintServer system you are using does not support tumble printing. If you requested both tumble and duplex printing and your PrintServer system does not support either option, you see only this error message.

User Action: Send your print job to a PrintServer system that supports tumble printing, or reissue the **lpr** command without specifying tumble printing.

Prologue *prologue name*, version *version number*.

Explanation: The prologue loader uses this message to find out what version of the ANSI prologue is loaded in the printer.

User Action: None.

'setpapertray' is not supported by multipage layup.

Explanation: The **setpapertray** operator is ignored when you use layup to print more than one page to a sheet.

User Action: None.

The current path may have been lost.

Explanation: You see this message when layup interacts with a POSTSCRIPT program that contains the results of the **charpath** operator in the current path, while executing a **showpage**.

User Action: Correct the POSTSCRIPT program. See the *Programmer's Supplement: POSTSCRIPT Printers* for information on POSTSCRIPT.

The upper limit of your job has been reached. The rest of your job will be flushed.

Explanation: This message is displayed when you use the `page_limit` option, to indicate that the number of pages you specified has been reached.

User Action: This is an informational message. No action is required if this is what you expected.

Unrecognized keyword 'keyword'

Explanation: The keyword does not have a recognized value.

User Action: Refer to Chapter 3 for information on acceptable values for *keyword*. Edit the layup definition file and reissue the **lpr** command.

Unrecognized option 'option'

Explanation: A line does not contain a recognized option.

User Action: Refer to Chapter 3 for information on acceptable options for layup definition files. Edit the layup definition file and reissue the **lpr** command.

Your job is not being printed because 'string' size paper was found in the 'string' paper tray when you specified 'string' size paper.

Explanation: This message is displayed when you specify both the `input_tray` and `sheet_size` options, but with conflicting values. For example, you would see this message if the bottom input tray contained A-size paper, but you specified the bottom input tray and sheet size B.

User Action: Put the paper size you require in the specified input tray and reissue the **lpr** command or reissue the command specifying an input tray that already has the paper size you require.

6.2 Accounting

Your system enables printer accounting only if a default accounting file has been specified by the **af** entry in the **/etc/printcap** file.

After a job has printed on a PrintServer system, the PrintServer Client for ULTRIX software obtains accounting information from the PrintServer distribution software. The line printer daemon routes the accounting record to the printer's accounting file, which has been specified in the **/etc/printcap** file.

To generate a report of your printer usage, use the **pac** command. For example:

```
# /etc/pac
```

The **pac** command displays a report detailing usage per user: number of pages printed, feet of paper consumed, and total estimated cost.

6.2.1 Job Flag Page

The job flag page is a one-page identification of the print job that is printed before the first page of the job. The job flag page contains the following items:

- User name and client node name
- Job identification number
- Job name (the name of the first file in the job)
- Printer queue name
- Date and time the job started printing
- The DIGITAL logo
- The ULTRIX logo
- The PrintServer product name

Figure 6-1 shows a sample job flag page that is printed at the beginning of a print job.

Figure 6-1 Sample Job Flag Page

root@rashi	
JOB 70	
client-chap.txt	
Printer queue:	script
Started:	Wed Jul 6 08:40:15 1989
Digital Equipment Corporation	
ULTRIX™	PrintServer 40

MLO-003441

Glossary

aspect ratio

Defines the shape of a piece of paper as the ratio of the vertical side of the sheet to the horizontal side.

bitmap

An image in digitized form that can be stored, transmitted, and reproduced precisely.

client

In the context of a PrintServer printer, any system running the client software. Users on that client system share the services of one or more PrintServer nodes in the network for remote printing.

client software

Installed on a VAX computer, client software handles users' print requests and provides direct communication with the printer.

configuration database

A set of values that describes an operating environment of the PrintServer system.

DECnet

A software implementation of the Digital Network Architecture protocol, which provides for the exchange of information between nodes in a local or wide area network.

device control library

In the context of the PrintServer Client for ULTRIX software, an ULTRIX text library file that contains modules of POSTSCRIPT procedures. The line printer daemon extracts the appropriate modules, depending on the option given in the **lpr** command line, and inserts the module at the head of the data stream in the print job.

down-line load

The process by which a host system sends a copy of an executable system image over a communications channel to another node in the network.

drawing primitives

In a PrintServer printer, bitmap area and character operations in halftones that the POSTSCRIPT interpreter creates and sends to the image controller. The image controller composes the image to be printed from these primitives.

duplex layout

A page layout for a document that is to be printed on two sides of a sheet. Page headings and page numbers often alternate right and left, so that when the sheet is printed they will always be on the outside edge of the page. In a document that has chapters, the first page of each chapter is a recto page, so duplex-formatted documents insert a blank page where necessary. Duplex documents are formatted for left binding. The wider margin for binding or hole-punching alternates, rather than remaining on the same side of the page.

duplex printing

Printing on both sides of a sheet. Documents printed using normal duplex are meant to be bound along the long edge of the sheet.

Ethernet address

A 48-bit value that uniquely identifies a node in an Ethernet network.

Ethernet network

A local area network that uses coaxial cable as a passive communications medium to interconnect computers, information-processing products, and office equipment at a local site without requiring switching logic or control by a central computer.

event logging

Recording information about significant events in the operation or maintenance of a network.

filter

A program that reads some input, transforms it, and writes some output.

font

The artistic representation of a typeface that describes a set of alphanumeric characters rendered in a particular point size, weight, and style.

font file

A data file that contains information used to reproduce a particular font.

generic queue

On a PrintServer printer, a queue that has a default data type associated with it.

host

A network node that performs services, such as down-line loading, for other nodes in the network.

landscape

An orientation in which lines of text are printed parallel to the long dimension of the paper.

laser electrophotography

A nonimpact printing technique in which a low-power laser generates an image one line at a time on the photoconductive surface of a drum.

layering

A POSTSCRIPT programming technique that allows device control modules and applications such as document formatting activities to redefine POSTSCRIPT operators in order to modify a print job's behavior.

layup

A Digital PrintServer feature that allows you to control the mapping of pages to sheets. For example, you can specify where a page will print on a sheet, or the number of pages that will print on a single sheet.

line printer daemon (lpd)

A spool area handler that receives requests to print files in the queue, display the queue, or remove jobs from the queue. The line printer daemon arranges to have print data translated if required, and queues the requests and data for transfer to the PrintServer printer.

Local Area Print Station (LAPS) protocol

The application level protocol used to provide the interface between client and PrintServer printer in a DECnet local area network environment.

local node

The node at which you are physically located.

local printer

In the context of the PrintServer Client for ULTRIX software, the printer to which your computer communicates directly. Your computer must have the PrintServer Client for ULTRIX software installed.

lock file

A file created by the line printer daemon (**lpd**) to reserve access to the printer for a particular print job. The presence of a lock file indicates that a job is printing.

network

A collection of interconnected individual computer systems.

node

An individual computer system in a network that can communicate with other computer systems in the network.

node address

The unique numeric identification of a specific node.

node identification

Either the node name or the node number of a network node.

node name

An optional alphanumeric identification associated with a node address in a one-to-one mapping. A node name must contain at least one alphabetic character.

number up

The term for printing multiple pages on one sheet of paper. For example, printing 4-up means that you are printing four pages on a single sheet.

page

In the context of a PrintServer printer, a page is the image you are printing; you usually print one page per sheet.

page spot

The place on a sheet that a page can be printed. Using layup, you can have multiple page spots on a single sheet.

point

Printer's unit of measure, used principally for designating type size. There are 12 points to a pica; approximately 72 points to an inch.

portrait

An orientation in which lines of text are printed parallel to the short dimension of the paper.

POSTSCRIPT

A programming language designed to convey a description of virtually any desired page to a printer. It can describe a page containing any combination of text, graphical shapes, and digitized images.

POSTSCRIPT interpreter

A stored program in the PrintServer printer that converts POSTSCRIPT commands into page descriptions.

primary supporting host

For a particular PrintServer system, the supporting host where event-logging data is currently being stored and where the configuration database is saved to or loaded from.

Printing Services Software

Software down-line loaded from a supporting host that runs on the PrintServer CPU. This software transfers print data from a client; formats the print data, using a built-in POSTSCRIPT interpreter; builds bitmaps; and sends the bitmaps to the data controller subsystem.

PrintServer controller

The hardware and software that processes the page descriptions from the POSTSCRIPT interpreter, builds bitmaps of each page to be printed, and forwards the bitmaps to the print engine to be converted to hardcopy output.

PrintServer distribution software

The software that handles the communications between the system that makes a print request (the client), the system that provides management resources (the supporting host), and the system that performs the printing service (the PrintServer printer). This software is distributed across the network such that some is executed on the PrintServer CPU, some is executed on the supporting host, and some is executed on the client systems.

PrintServer printer

A network laser output device that contains software to receive incoming print files, a controller to process POSTSCRIPT page descriptions into bitmap images, and a print engine to produce marks on paper or transparency.

printing system

A group of components that provide network printing services. A printing system consists of a client, supporting host, and printer.

recto

An odd-numbered, right-hand page in a book. For example, page 1 is always a recto page. For a single sheet, the recto page is the page on the front of the sheet. *See also* **verso**.

remote node

To the local node, any other network node.

remote printer

To the local printer, any other network printer.

resource

An identifiable entity, physical or conceptual, that can perform a service and can be named and accessed on the network. The shareable entity can be either hardware or software.

server

A node dedicated to performing a specific service or set of related services, for example, a PrintServer network printer.

sheet

In the context of a PrintServer printer, a sheet is the physical piece of paper that is printed.

simplex layout

A page layout for a document designed to be printed on one side of a sheet. The page headings and page numbers of a simplex document appear in the same position on each page. The margins remain the same for every page. For example, if the document is to be put in a binder, every page will have a wide left margin for hole-punching.

simplex printing

Printing on one side of a sheet.

spooling directory

In the context of ULTRIX printing, the directory in which the **lpr** command stores files waiting to be printed.

supporting host

In the context of a PrintServer system, a VAX computer, running supporting host software. A supporting host system can down-line load a PrintServer printer and provide it with management services.

TCP/IP

The generic name for a group of network and transport layer network protocols used to move data between computers.

translator

In the context of a PrintServer system, a stored program that translates a particular imaging data syntax, such as ANSI or ReGIS, into the POSTSCRIPT page description language.

tumble layout

A page layout for documents that are meant to be tumble printed. The page headings may alternate position. However, in tumble format, rather than alternating right and left, recto pages may have the page headings at the bottom of the page, and verso pages may have the page heading at the top. The margin left for hole-punching alternates along the two short edges of the sheet. Recto pages are printed on the first side of the sheet, so blank pages are inserted where necessary.

tumble printing

Printing in which pages are printed on two sides of a sheet, and alternate pages are rotated 180 degrees. Tumble-printed documents are designed to be bound on the short, rather than long, edge. For example, a tumble document may be bound at the top of each sheet, so you turn each page from the bottom. Legal documents are often tumble printed.

user

A person (or possibly a program) on a client or supporting host system that initiates print requests for a remote PrintServer printer.

verso

An even-numbered, left-hand page in a book. For example, page 2 is always a verso page. For a single sheet, the verso page is the page on the back of the sheet. *See also* **recto**.

write-black

The laser imaging technology used in the PrintServer 20 model to produce black marks on print media, such as paper or transparencies. In this technology, a low-power laser generates an image one line at a time on the photoconductive surface of a negatively charged Organic Photo Conductor (OPC) drum. The light discharges those portions of the drum on which an image is to appear. When the printer applies a dry powder (toner) to the drum, the negatively charged toner particles adhere to the discharged areas of the drum. The toner is first transferred to the media; then when it is pressed against the media and heated, the powder is fused to the media. *See also* **write-white**.

write-white

The laser imaging technology used in the PrintServer 40 model to produce black marks on print media, such as paper or transparencies. In this technology, a low-power laser generates an image one line at a time on the photoconductive surface of a positively charged selenium drum. The light discharges those portions of the drum on which no image is to appear. When the printer applies a dry powder (toner) to the drum, the negatively charged toner particles adhere to the positively charged areas of the drum. The toner is first transferred to the media; then when it is pressed against the media and heated, the powder is fused to the media. *See also* **write-black**.

C

Client
 definition of, 1-2
Console
 PrintServer 40, 1-13

D

Data type
 See **lpr** PrintServer options, *data_type*
DEQNA
 See PrintServer 40 hardware components
Device control library, 1-4, 4-1
Device control modules, 4-1
 creating, 4-3
 lpr PrintServer options, 4-1
 PostScript extensions, 4-2
 redefining PostScript operators, 4-2
Drilled paper, 5-1
Duplex, 2-16

E

Error messages
 See Messages

G

Gummed labels, 5-2

I

Input tray
 See **lpr** PrintServer options, *input_tray*
iplpscomm
 description, 1-4

L

Labels, 5-2
Landscape orientation
 See **lpr** PrintServer options, *orientation*
LAPS protocol, 6-4
Layup, 3-1
 See PrintServer options, *layup*

Layup definition file
 creating, 3-1
 error messages, 6-10
Letterhead paper, 5-1
Line printer daemon (**lpd**)
 definition of, 1-3
lpc command, 2-24
lpq command, 2-23
lpr PrintServer options, 2-2
 data_type, 2-3
 device control modules, 4-1
 error messages, 2-21
 flags for, 2-3
 , 2-4
 layup, 2-5
 message, 2-6
 number_up, 2-6, 3-6, 3-8
 orientation, 2-9
 output_tray, 2-10
 page_limit, 2-13
 page_size, 2-13
 rules for, 2-2
 sheet count, 2-14
 sheet_size, 2-14
 sides, 2-15
lpr syntax, 2-2
lprm command, 2-24
lpscomm
 description, 1-4

M

Media
 drilled paper, 5-1
 gummed labels, 5-2
 input and output devices, 1-9
 input and output trays, 1-12
 letterhead paper, 5-1
 preprinted forms, 5-2
 transparencies, 5-2
Messages
 descriptions, 6-1
 format, 6-1
 lpr command, 2-21
 lpr PrintServer option, 2-6
 operator, 6-1
 user, 6-1

N

Number up

See **lpr** PrintServer options, number_up

O

Orientation

See **lpr** PrintServer options, orientation

Output tray

See **lpr** PrintServer options, output_tray

P

Page limit

See **lpr** PrintServer options, page_limit

Page mapping

See **lpr** PrintServer options, layup

Page size

See **lpr** PrintServer options, page_size

Paper

input and output devices, 1–9, 2–4, 2–10

input and output trays, 1–12

Portrait orientation

See **lpr** PrintServer options, orientation

Preprinted forms, 5–2

Print queue commands, 2–23

Printing Services Software, 1–13

Printing system

definition of, 1–1

PrintServer 20

controller, 1–9

hardware description, 1–7

print engine, 1–8

print speed, 1–8

selectable print engine features in, 1–8

PrintServer 40 series

data controller, 1–12

hardware description, 1–10

image controller, 1–13

print engine, 1–11

print speed, 1–11

PrintServer 40 series hardware components

console, 1–13

DEQNA, 1–13

device controller, 1–13

PrintServer family, 1–6

PrintServer models

comparison of, 1–6

hardware components in, 1–10

Punched paper

See Drilled paper

R

Recto page, 2–15

S

Sheet count

See **lpr** PrintServer options, sheet_count

Sheet size

See **lpr** PrintServer options, sheet_size

Sides

See **lpr** PrintServer options, sides

Simplex, 2–16

Supporting host, definition of, 1–5

T

Translators

definition of, 1–4

using with **lpr** command, 2–3

Transparencies, 5–2

Tumble layout, 2–18

Tumble printing, 2–19

U

ULTRIX print queue commands, 2–23

V

Verso page, 2–16

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local Digital subsidiary or approved distributor
Internal*	_____	SSB Order Processing - WMO/E15 <i>or</i> Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to PrintServer Clients
AA-PC31B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
------	-------------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

----- Do Not Tear - Fold Here and Tape -----

digitalTM



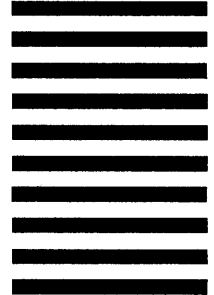
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK ROAD
NASHUA NH 03062-2698



----- Do Not Tear - Fold Here -----

Cut
Along
Dotted
Line