Advanced Printing Software

User Guide

September 2002

Product Version: Advanced Printing Software Version 1.2

Operating System and Version: Tru64 UNIX Version 5.1B or higher.

This guide contains information for users who want to submit and manage print jobs using the Advanced Printing Software command-line interface.

Hewlett-Packard Company
Palo Alto, California

Advanced Printing Software is based on Xerox PrintXchange technology developed and marketed by Xerox Corporation.

Contents

3 Printing Your Job

4 Managing Your Print Job

A Attribute Reference

B BSD Print System Command Equivalents

Index

Tables

About This Manual

The *Advanced Printing Software User manual* provides information on submitting and monitoring jobs to your printer using the command-line interface (CLI) on your workstation.

Audience

This manual is intended for all users who need to submit and monitor print jobs using the Advanced Printing Software command line-interface.

Related Software

Advanced Printing Software features PrintXchange technology developed and marketed by Xerox Corporation.

Organization

This manual is organized as follows:

- *Chapter 1* – An introduction to the manual and an overview of the Advanced Printing Software
- *Chapter 2* – Information on defining and using logical printers, examining printer properties and settings, and setting up printer attribute files
- *Chapter 3* – Step-by-step instructions for submitting a print job
- *Chapter 4* – Guidelines for managing jobs that are waiting to print
- *Appendix A* – Summaries of command attributes for jobs, documents, and printers
- *Appendix B* – A comparison of BSD print commands to Advanced Printing Software commands

Related Documentation

In addition to this manual, the following publications are available:

- *Advanced Printing Software Command Reference Guide*. A guide that provides information on using, maintaining and operating the print system using the CLI.
- *Advanced Printing Software System Administration and Operations Guide*. A guide explaining installation, configuration, and day-to-day

management of the print system. It is primarily intended for administrator and operator use.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This manual uses the following typographical and symbol conventions:

%	A percent sign represents the C shell system prompt.
$	A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
% **cat**	Boldface type in interactive examples indicates typed user input.
file	Italic (slanted) type indicates variable values, placeholders, and function argument names.

[|] { | }	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
`cat`(1)	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.

1

Introduction

Advanced Printing Software is a network printing service. It manages jobs and printers from anywhere on the local network regardless of type of printer. It provides a smart link between you and the available printers.

1.1 Benefits of Using Advanced Printing Software

When you use printers managed by Advanced Printing Software, you gain several benefits. These include:

- Submitting jobs to logical printers instead of a specific hardware device. The job you submit to a logical printer is validated to make sure a physical printer exists to print the job correctly. The print system directs your job to the appropriate physical printer if one exists or rejects your job if one does not exist.
- Sending your print jobs to any of the available printers without using complicated setup procedures. All you need to know is the physical location of the printers.
- Monitoring the status of your print job. After you send your job to a printer, you can monitor its progress using either a graphical user interface (GUI) or command-line interface (CLI).
- Receiving notification of your print job's status. This includes notification that a particular printer is not in service or that your print job is completed.
- Obtaining improved administrator support. By using the integrated printer and job management tools, your administrator can better maintain the print system.
- Interoperating with a legacy BSD-based printing system. Advanced Printing Software accepts print requests issued by `lpr` or `lp` commands and provides the basic functionality required to process them. It can also forward jobs to remote lpd servers and printers.

1.2 Advanced Printing Software Terminology

The following terms describe how Advanced Printing Software processes your print requests.

- **Document** – A single file to be printed.

- **Job** – a collection of one or more documents that are to be printed as a unit.
- **Server** – A system process that provides print services. There are two kinds of servers: spoolers and supervisors.
- **Logical printer** – A software representation of one or more physical printers. You direct your print request to a logical printer whose characteristics fit the needs of the job. Logical printer objects reside in spooler processes and their databases.
- **Physical printer** – A software representation of a physical device. Physical printer objects reside in supervisor processes and their databases.
- **Physical device** – An actual output device with specific characteristics and capabilities. Examples include an HP LaserJetIII or a Xerox 4230 laser printer.
- **Queue** – A queue serves as a pool that holds jobs until they are ready to print. Logical printers insert jobs into a queue; physical printers take jobs from the queue.
- **Object** – An abstraction used to represent various entities, such as printers and queues. Each print system object contains a collection of attributes.
- **Attributes** – Characteristics of an object relating to its identity, physical makeup, or status. Every print system object contains a collection of attributes that provide information about that object. For example, the `printer-state` attribute indicates the current state of a printer, such as *idle* or *printing*. When you submit a job, you can specify attributes for your job and the documents it contains.

1.3 Access to the User Interfaces

In addition to the command-line interface described in this manual, the Advanced Printing Software contains a graphical user interface (GUI) that can be used to submit print jobs and monitor print jobs.

The pdprint GUI program submits print jobs. The pdprintinfo GUI program obtains job and printer status. These GUIs are accessible from the command-line or from the Print Manager icon of the CDE desktop. The following methods can be used to access these programs:

- The pdprint program can be accessed by entering the command on the command-line or by dragging and dropping a file that you want printed onto the Print Manager icon on the CDE desktop.

- The pdprintinfo can be accessed by entering the command on the command-line or by selecting Print Manager from the Personal Printers dropdown menu.
- The pdprint and pdprintadmin GUIs can also be accessed by selecting Advanced_Printing from the Application Manager.

1.4 Command Syntax Elements

The syntax for all print system commands has the following format:

```
command option option-argument object-instance
```

The syntax elements are:

- Command name
- Option
- Option argument
- Object instance

The following is an example of the complete command-line:

```
pdpr -n 2 opus1.txt
```

1.4.1 Specifying Options and Arguments

Options and option arguments modify the default behavior of print system commands. The following guidelines apply to using options and arguments with the CLI commands:

- Option names consist of a minus sign and a single lowercase or uppercase letter; for example, `-h`.
- All options must precede the operand.
- An option argument must follow the option, separated by a space. When an option includes several arguments, each argument must be separated from other command elements by a space.
- Commands interpret options and option arguments in the order in which they appear on the command-line.
- Not all options work with all commands. Refer to the *Advanced Printing Software Command Reference Guide* for a detailed description of each command to determine which options are valid for that command.

1.4.2 Specifying Operands and Classes

Some commands require you to specify at least one operand. A command operand specifies an object such as a file name on which you want to perform an operation.

Several commands that take operands can perform an operation on different classes of print system objects. The class of an object indicates what kind of object it is, such as a printer, queue, job, server, or document. Use the `-c` *`class`* option to specify the operand class of a command.

The following table summarizes the commands available to end users.

Table 1–1: End User Commands

Command	Description
pdls	Lists print object attributes.
pdmod	Modifies previously submitted print jobs or documents.
pdpause	Pauses your print jobs.
pdpr	Submits a print job.
pdq	Reports on or obtains status of print jobs.
pdresubmit	Resubmits print jobs.
pdresume	Resumes your own print jobs.
pdrm	Removes (that is, cancels) your print jobs.

1.5 Attribute Guidelines

The print system manages print jobs by using objects such as printers, jobs, documents, and queues. Every object has attributes and associated attribute values. For example:

- Job and document attributes determine the printer requirements of jobs and documents.
- Text job attributes provide controls for printing simple text jobs.
- Physical printer attributes define the capabilities of the printer device the object represents.

The following sections provide guidelines for using attributes with print commands.

1.5.1 Modifying Default Attribute Values

Some attributes have default values. However, for most attributes, the default value is no value. Attribute values can be modified in several ways.

- To override an attribute's default value, use the `-x` and `-X` options to specify a different value.
- To change an attribute value to its default value, use the `pdmod` or `pdset` command and specify the attribute name followed by `= =` with no value.

For a detailed directory of all attributes and their associated values, refer to the *Advanced Printing Software Command Reference Guide*.

1.5.2 Attribute Value Syntax

Some attributes can have only one value at a time (singlevalued). Others can have multiple values (multivalued). Yet other attributes can have one or more values, where each value itself has multiple components (complex).

This section describes the syntax for single-valued, multivalued, and complex attributes.

1.5.2.1 Syntax for Single-Value Attributes

The syntax for single-valued attributes is:

```
attribute=value
```

For example:

```
copy-count=2
```

If an attribute value is a text string with embedded space characters, you must enclose the string inside two sets of quotes. For example:

```
job-comment='"Deliver to Joe"'
```

1.5.2.2 Syntax for Multivalue Attributes

The syntax for multivalued attributes is:

```
attribute="value_1 value_2 ...  value_n"
```

For example:

```
content-orientations-supported="portrait landscape"
```

1.5.2.3 Syntax for Complex Attributes

The syntax for complex attributes is:

```
attribute="{attribute=value attribute=value at-
tribute=value}"
```

For example:

```
access-control-list="{name-type=all-users privilege-level=end-user}"
```

1.5.3 Abbreviating Attributes and Values

You can abbreviate attributes and standard identifier values by using only a few letters of each word in the name or value. For example, you can use the abbreviation `j-s` for the `job-sheets` attribute, `j-c-s` for the `job-copy-start` value, and specify the attribute as `j-s=j-c-s`.

The system accepts only unambiguous abbreviations. For example, abbreviating `job-owner` as `j-o` is not valid because it can also stand for `job-originator`. You need to specify enough of the attribute or value name so that it is unique. Examples of valid abbreviations are as follows:

`j-ow` or `job-own` for `job-owner`
`j-or` or `job-orig` for `job-originator`
`i-a3-w` for `iso-a3-white`

You cannot abbreviate name values that are not standard identifiers, such as site-specific media or tray names.

1.6 Job and Document Identifiers

Every print job is identified by a unique job identifier, comprised of a server (spooler) name, a colon, and a job number. The server assigns a job identifier when it accepts your job as part of a print request.

Examples of valid job identifiers are galileo_spl:1564 and kepler:1571.

Several commands, such as `pdpause` and `pdresume`, accept job identifiers as operands. Such commands also allow you to specify a document identifier in a multidocument job.

Some commands require you to identify a specific document within a multidocument job. A document identifier is expressed as a job identifier, a period, and a document number. Within each multidocument job, the documents are numbered starting with 1.

If kepler:1571 is a valid job identifier, then the second document in that job would be kepler:1571.2.

1.7 Getting Help for Commands

To get help for a command and its options, enter the command name followed by the `-h` option. For example, for help on the `pdpr` command, enter:

```
% pdpr -h
```

You can also use the `man` command to display information about the print system commands. For example, for help on the `pdpr` command, type:

```
% man pdpr
```

2

Defining Printer Defaults

This chapter explains how to define and use a logical printer, examine printer properties and settings, and set up printer attribute files. A logical printer is a software representation created by the system administrator for one or more physical printers. For example, one logical printer might represent three diverse printers: one that prints on standard paper, one that prints with highlight colors and stapling, and another that prints on large media. You send a print request to a logical printer whose characteristics fit your needs and the server determines if and how it can print that job.

2.1 Defining the Name Service

If your print system uses LDAP as a name service, you need to edit or create the `/var/pd/config/apx.conf` file. The content of this file should be obtained from the print system administrator.

The `/var/pd/config/apx.conf` file contains information on name services you use and contains information about the host system that is running the LDAP directory services. The following example shows the contents of a `/var/pd/config/apx.conf` file:

```
name-services = file nis ldap
LDAP_hosts = system.abc.xyz.com
LDAP_path = ou=organizational unit,o=organization
```

- `name-services` describes the name services the print system uses and the order in which they are used. In this example, local file, NIS, and LDAP are used. Lookups for print objects are done in the local file (`/etc/printers.conf`), in the `printers.conf.byname` NIS map, and lastly in the LDAP directory server. The default name services are file and nis.
- `LDAP_hosts` lists up to three hosts that the LDAP directory services is running on. Multiple hosts are listed when they are providing replication services. When you list multiple hosts, they must be separated by colons. If the LDAP server is on a system running TruCluster Version 5.0 or later software, you can use the cluster alias as the host name.

  The standard LDAP services port number is 389. However, LDAP services can be provided on a nonstandard port number. You specify the LDAP port number by appending the port number to

the host name within parenthesses as in the following example:
`LDAP_hosts = myhost.xyz.com(8182)`

- `LDAP_path` lists a directory search path. To access an LDAP-based database, you must supply a distinguished name path. Printing clients and servers use this path to create and search for entries in the directory. With Netscape Directory Server, you start by naming your organization, and then you create organizational units under the organization. The distinguished name path you need to specify for Advanced Printing includes the organization and organizational unit in distinguished name syntax. For example, if your top-level organization is named XYZ Corp, and you define an organizational unit AdvPrint, the distinguished name path would be:

  `ou=AdvPrint,o=XYZ Corp`

  You can specify up to three entries in the `LDAP_path` expression. Separate multiple entries with a colon (:) character. For example, if in XYZ Corp, you create a secondary organizational unit called Test Environment, your `LDAP_path` should be specified as follows:

  `ou=AdvPrint,o=XYZ Corp:ou=Test Environment,o=XYZ Corp`

  When accessing entries listed in the `LDAP_path` expression, a search continues through the entries until a matching entry is found. When creating or deleting entries, only the first path entry is used; the second and third entries listed in the `LDAP_path` expression are read-only.

2.2 Defining Your Default Logical Printer

If you submit all or most of your jobs to the same logical printer, you can define it to be your default printer. This also reduces the need to specify a logical printer in many print system commands.

Consider the following scenario:

- Printer *same_old* prints jobs in black and white.

  Because you print most often in black and white on regular-sized paper, you can specify this as your default logical printer.

- Printer *flashy* prints in highlight colors and can staple documents.

  Occasionally, you need to print using highlight colors, so you can override the default setting and specify this logical printer.

- Printer *jumbo* prints in black and white on oversized paper.

  Sometimes you need to print on oversized paper in black and white. If so, you can override the default setting and specify this logical printer.

The PDPRINTER environment variable, which you can define in your shell or in your `.profile` or `.login` file, determines your default logical printer.

To set or change the default logical printer, use one of the following procedures, substituting the name of your default printer for *PrinterName*.

- Procedure for C shell users:

  1. Insert or modify the following line in the `.login` file in your home directory and save the file:

     $ **setenv PDPRINTER** *PrinterName*

  2. Apply the changes to the `.login` file by entering the following command:

     $ **source .login**

  3. Display the name of the logical printer by entering this command:

     $ **echo $PDPRINTER**

- Procedure for Korn shell users:

  1. Insert or modify this line in the `.profile` file in your home directory and save the file:

     $ **export PDPRINTER=***PrinterName*

  2. Apply the changes to the `.profile` file by entering this command:

     $ **. ./.profile**

  3. Display the name of the logical printer by entering this command:

     $ **echo $PDPRINTER**

After you set the PDPRINTER environment variable, any print job you submit without designating a printer is inserted into the default logical printer's queue. In addition, the spooler server associated with your default logical printer becomes your default server. When you issue certain commands without specifying an operand, the system uses the default server value to provide a useful result in the context of that server.

2.2.1 Defining a Default Logical Printer for the CDE Mail Tool

If you have not specified a value for PDPRINTER environment variable in your `.dtprofile` file, and you print from the CDE mail tool, a dialog box is displayed stating that you have not specified a default printer. Printing succeeds if you specify a working logical printer.

To set the default logical printer, insert the following line in your `.dtprofile` in your home directory:

export PDPRINTER=<printer name>

Save the file.

2.3 Accessing the Default Logical Printer

To specify a default logical printer, use the `pdpr` command to submit a print job. For example, to submit the file `budget99.txt` to the default logical printer, enter:

```
$ pdpr budget99.txt
```

You can use the `pdq` command to return a list of your jobs on the default printer. To display such a list, enter:

```
$ pdq
```

2.4 Accessing a Specific Logical Printer

To specify a logical printer, use the `-p` *`printer_name`* option with the `pdpr` command. For example, to submit the file `budget99.txt` to the highlight color printer named *flashy*, enter:

```
$ pdpr -p flashy budget99.txt
```

Use the `-p` option with the `pdq` command to return a list of jobs you have submitted to a specific logical printer. For example, to display a list of jobs you have submitted to logical printer flashy, enter:

```
$ pdq -p flashy
```

To find the names of all available logical printers, use the `pdls` command as follows:

```
$ pdls -c printer
```

2.5 Listing Logical Printer Attribute Properties

You may have a variety of printers available to you in your networked environment. To determine which printer best meets your needs, use the `pdls` command to list printer properties, including:

- The printer name
- Whether the printer can accept print requests
- Supported features such as input trays, output bins, media, and native document formats
- How many jobs are pending in the print queue for the printer

You can request a list of all available logical printers supported by your default server by entering the following command:

```
$ pdls -c printer
```

To request information about all printers on a particular spooler, specify the server name followed by a colon. For example, to find out which printers are supported by the spooler doggone_spl, issue the following command:

```
$ pdls -c printer doggone_spl:
```

The following example requests basic information about the printer bulldog:

```
$ pdls -c printer bulldog
```

To request an expanded list of attributes that includes the associated server and printers, use the `-r verbose` option:

```
$ pdls -c printer -r verbose bulldog
```

To request a complete list of all attributes including printer features, use the `-r all` option and the `-s line` option. The latter option indicates that you want the output displayed one attribute per line instead of wrapping to the width of your window, which is harder to read.

```
$ pdls -c printer -r all -s line bulldog
```

2.5.1 Requesting Specific Printer Attributes

You can request specific printer attributes by including the `-r` option with the `pdls` command. For example, the `printer-name` attribute stores the names of logical printers associated with a server and the `printer-associated-printers` attribute stores the names of physical printers associated with a logical printer. The following example shows how to determine the logical and physical printers available on server bulldog_spl.

```
$ pdls -c printer -r \
  "printer-name printer-associated-printers" bulldog_spl:

printer-name  printer-associated-printers
------------  --------------------------
fetch         bone_pp
              stick_pp
              slipper_pp
two_sidedPS   ln1701_pp
              ln1702_pp
biglab        office_pp
              hallway_pp
              javaroom_pp
              closet_pp
```

2.6 Setting Default Logical Printer Attributes

When you issue print requests, you probably use certain attributes frequently, such as specifying job start sheets. Rather than repeating the

same attributes with every print request, you can apply default settings by using `initial-value-job` and `initial-value-document` objects.

An `initial-value-job` object is a set of attributes and values stored on the server. The set of attributes is applied to a job all at once, when you specify the object while submitting a job, or automatically whenever a job is directed to a particular logical printer.

An `initial-value-document` object is a similar set of attributes and values, except that it applies to documents rather than jobs.

Because every print system configuration is unique, you should check with your administrator for details on the use of such objects. If you know the name of the spooler server that manages your logical printers, you can use the `pdls` command to ask the system for its initial value objects.

For example, the following command lists all `initial-value-job` objects on server doggone_spl:

```
$ pdls -c initial-value-job doggone_spl:
```

If the result of the preceding example included an object named bulldog_IVJ_DEFAULT, you can list the details of that object by using the following command:

```
$ pdls -c initial-value-job -r all -s line \
doggone_spl:bulldog_IVJ_DEFAULT
```

The result lists those attributes that are applied to a job whenever that `initial-value-job` object is either specified with a job or associated with the printer.

2.6.1 Setting Default Print Job Attributes

Your print system administrator can apply default settings to all jobs submitted to a printer by associating an `initial-value-job` object with a logical printer's `printer-initial-value-job` attribute. The print system uses this attribute to apply attribute values to all jobs submitted to the printer.

For example, a printer's `initial-value-job` object can include the `job-sheets=job-copy-start` attribute. All jobs you submit to that logical printer include, by default, job start sheets.

You can also specify `initial-value-job` objects when you submit a print job.

2.6.2 Setting Default Print Document Attributes

Your administrator can apply default attribute values to every document submitted to a printer. In this case, an `initial-value-document` object is associated with a logical printer's `printer-initial-value-document` attribute.

For example, a printer's `initial-value-document` object might include the `copy-count=2` attribute. As a result, the printer makes two copies of every document in every job.

When you submit a print jo, you can also specify `initial-value-document` objects, such as one that specifies a different number of copies.

2.6.3 Checking a Logical Printer for Initial-Value Objects

To determine the name of the `initial-value-job` and `initial-value-document` objects associated with the printer bulldog, use the following command:

```
$ pdls -c printer -r \
  "printer-name printer-initial-value-job printer-init-val-doc" \
  bulldog
```

Then, assuming that the initial-value objects are `bulldog_IVJ_DEFAULT` and `bulldog_IVD_DEFAULT` respectively, and that the spooler name is doggone_spl, display the attributes and values represented by the initial-value objects using the following commands:

```
$ pdls -c initial-value-job -r all \
-s line doggone_spl:bulldog_IVJ_DEFAULT

% pdls -c initial-value-document \
-r all -s line doggone_spl:bulldog_IVD_DEFAULT
```

The resulting output lists attributes that are defined by the initial-value objects. For example, the command to list `initial-value-job` information may produce the following:

```
bulldog_IVJ_DEFAULT: object-class         = initial-value-job
bulldog_IVJ_DEFAULT: initial-value-job-identifier = bulldog_IVJ_DEFAULT
bulldog_IVJ_DEFAULT: job-hold             = no
bulldog_IVJ_DEFAULT: job-retention-period = 2:00
bulldog_IVJ_DEFAULT: associated-server    = dogear_spl
bulldog_IVJ_DEFAULT: job-sheets           = job-copy-start
```

2.6.4 Using an Attribute File

You can predefine specific attributes in text files called attribute files. You can then include them with a printer command by using the -X option to specify an attribute file. Refer to the *Advanced Printing Software Command Reference Guide* for more information.

In the following example, an attribute file called `budget_format.attr`, located in your current working directory, contains a set of attributes for printing PostScript documents two-sided with no document sheets.

```
# Attribute file budget_format.attr
# Use this for printing 2-sided PostScript documents
document-format=PostScript
document-sheets=none
sides=2
```

The following example uses that attribute file to print the document budget99.ps as a PostScript file, double-sided, and without document sheets:

```
$ pdpr -X "budget_format.attr" budget99.ps
```

3

Printing Your Job

This chapter explains how to submit jobs to a printer and how to specify characteristics for printed output by using attributes. For a detailed explanation of attributes and summaries of all available attributes, see Appendix A.

3.1 Submitting a Print Job

Use the `pdpr` command to print a job. For example, the following command submits a document file called `report.txt` to the default printer:

```
$ pdpr report.txt
```

The following command line sends three files to the default printer by using a space to separate the file names:

```
$ pdpr report.txt table.txt chart.ps
```

To indicate a specific printer, use the `-p` option and its `printer_name` argument. Always place the options and arguments before the document file names they apply to.

The following command line submits `report.txt` and `table.txt` to the printer named pawprint:

```
$ pdpr -p pawprint report.txt table.txt
```

3.2 Specifying Job Attributes

You can specify different characteristics for each job and document by using attributes. Enter a value for an attribute by using the `-x` option. For example, to print a two-sided copy of `budget99.txt`, enter the following:

```
$ pdpr -x "sides=2" budget99.txt
```

Note

In some examples, quotation marks are shown around single attributes. However, quotation marks must surround multiple attributes or an attribute value containing spaces.

To specify more than one attribute in a print request:

- Bundle them together inside quotes, following the `-x` option; you can specify each attribute with its own `-x` option
- Store the attributes in an attribute file and specify that file with the `-X` option.

The following commands are equivalent:

```
$ pdpr -x "sides=2 job-comment='After the takeover'" budget99.txt
```

```
$ pdpr -x sides=2 \
-x job-comment="'After the takeover'" budget99.txt
```

3.2.1 Setting Print Job Priority

You can set the job priority of a new print job or you can change the priority of a print job that has been submitted.

To set the priority of new print job, `new_job.txt` to 20, use the `pdpr` command with the `job-priority` attribute, as shown in the following example:

```
$ pdpr -x job-priority=20 new_job.txt
```

To change the priority of a print job that has been submitted, first use the `pdq` command to get the job identification number. Then, use the `pdmod` command with the `job-priority` attribute to change the job's priority. For example, to change the priority of an existing print job identified as 0102 to a priority of 25, enter the following:

```
$ pdq
job-identifier  job-name     current-job-state intervening-jobs
--------------  --------     ----------------- ----------------
dogear_spl:0102 budget02.txt pending           0
% pdmod -x job-priority=25 dogear_spl:0102
```

3.2.2 Specifying the Number of Copies

You can print multiple copies of individual documents or of an entire job. To specify the number of copies in a job, use the pdpr command with the `job-copies` attribute or with the `-n` option. The system prints one copy by default.

To print two copies of `budget99.txt` and `inventory99.txt` on the default logical printer, use either of the following examples:

```
$ pdpr -x "job-copies=2" budget99.txt inventory99.txt
```

```
$ pdpr -n 2 budget99.txt inventory99.txt
```

To print a different number of copies for each document in a job, use the `copy-count` attribute. The following example prints two copies of the file `slides.ps` and three copies of the file `handout.ps`.

```
$ pdpr -x copy-count=2 slides.ps -x copy-count=3 handout.ps
```

3.2.3 Specifying Two-Sided Printing

If your environment includes printers that can print on both sides of a sheet of paper, you can specify one-sided or two-sided printing with the `sides` attribute.

To determine which printers offer two-sided printing, use the following `pdls` command line:

```
$ pdls -c printer -r "printer-name sides-supported"
printer-name  sides-supported
------------  ---------------
bulldog       1
              2
boxer         1
```

The output shows that the only the first printer supports two-sided printing. To print a two-sided copy of the file `budget99.txt`, enter the following:

```
$ pdpr -p bulldog -x sides=2 budget99.txt
```

If the printer does not have a value for `sides-supported`, you cannot use the `sides` attribute for your job. The printer prints the document according to control instructions within the document file, or according to the default sides value set on the printer device front panel.

Some documents, particularly those created by PC-based applications, include printer instructions that force them to print as one-sided or two-sided, regardless of how you specify the print command.

3.2.4 Specifying the Document Format

The print system can print jobs in document formats such as text, PCL, and PostScript. When you submit a print job, you usually do not need to specify a document format. The print system determines the format by inspecting a small portion of each document.

There may be instances where the print system cannot determine the document format and the result may be unacceptable. In this case, you should specify the format with the `document-format` attribute.

To print a file called `review` on the default logical printer and specify PCL as the document format, enter:

```
$ pdpr -x "document-format=PCL" review
```

3.2.5 Specifying Media

In some print environments, media types are associated with input trays by the operator or administrator who assigns values to the `input-trays-medium` attribute. If this is the case, you can select a medium for printing by using the `default-medium` attribute. You can ask your administrator for the specifics of your configuration, or you can use the pdls command with the `media-supported` attribute. For example, to determine what media is supported for logical printer bulldog, enter:

```
$ pdls -c printer -r media-supported bulldog
```

If the command reports nothing, then you need to specify an input tray to select media.

Before you submit a job, you should first determine if the `input-trays-medium` attribute is defined on the physical printer. For example, if your physical printer is bulldog1_pp, then you can enter the following command line to view the `input-trays-medium` attribute:

```
$ pdls -c printer -r input-trays-medium bulldog1_pp
```

To submit a file called `mailer` to printer bulldog and specify `monarch-envelope` as the medium to use, enter:

```
$ pdpr -p bulldog -x "default-medium=monarch-envelope" mailer
```

If the document contains tray-selection operators that override the printer default settings, then unexpected media selection can occur.

3.2.6 Specifying Input Trays

Another way of selecting printing media is to use the `pdpr` command with the `default-input-tray` attribute. First, determine the input trays supported by your printer, as in the following example:

```
$ pdls -c printer -r input-trays-supported pawprint
input-trays-supported
---------------------
top
bottom
large-capacity
```

To print the file called `cad23.ps` on the printer pawprint using the bottom input tray, enter:

```
$ pdpr -p pawprint -x "default-input-tray=bottom" cad23.ps
```

If the document contains tray-selection operators that override the printer default settings, then unexpected media selection can occur.

3.2.7 Specifying Output Bins

You can specify the output bin for your print job by using the `pdpr` command with the `output-bin` attribute.

First, determine the output bins available for your printer as in the following example:

```
$ pdls -c printer -r output-bins-supported bulldog
output-bins-supported
---------------------
top
side
```

To print the file `budget99.txt` and place it in the side output bin, enter:

```
$ pdpr -x output-bin=side budget99.txt
```

3.2.8 Specifying Finishing Processes

Some printers offer advanced hardware options that perform finishing operations, such as trimming, binding, and stapling. To determine if any printers support finishing operations, use the following command line:

```
$ pdls -c printer -r "finishings-supported"
```

To print a job using a finishing process, include the `finishing` attribute in the pdpr command. For example, to print a saddle-stitched copy of `budget99.txt`, enter:

```
$ pdpr -x "finishing=saddle-stitch" budget99.txt
```

3.2.9 Printing from Standard Input

You can print jobs that originate from standard system utilities by using the shell's pipe capability. For example, to print a listing of files in your current directory, you can pipe the output of the `ls` command directly to your default printer:

```
% ls -l | pdpr
```

3.3 Specifying Job Scheduling Attributes

When you submit a print job, it is scheduled on a first in/first-out basis. Resource checking ensures that jobs are delivered only to printers that can print the job.

In some instances, you might want to submit a job with instructions to print at a later date. For example, you might want to schedule a long job to print after hours. You might also want to set a retention period for the job so you can check its status after you return to work.

The following sections explain job scheduling attributes that allow you to control when jobs are printed and how long they are retained.

3.3.1 Holding a Print Job

The `job-hold` attribute specifies whether a print job is to be printed or put on hold. The default value is `no` (do not hold).

If you have submitted a job and it is still waiting in its print queue, you can use the `pdmod` command to set its `job-hold` attribute to `yes` to put it on hold. The spooler server then prevents the job from printing. Other jobs submitted to the same printer (but not on hold) will continue to be scheduled normally. The held job will remain unprinted until you set the `job-hold` attribute to `no` or until the time specified by `job-discard-time` arrives.

The following example uses the `pdmod` command to put job dogear_spl:1027 on hold:

```
$ pdmod -x "job-hold=yes" dogear_spl:1027
```

At a later time, you can send the job to print with the following command line:

```
$ pdmod -x "job-hold=no" dogear_spl:1027
```

Note

You can use the `pdpause` and `pdresume` commands independently of the value of the `job-hold` attribute.

3.3.2 Printing a Job After a Specific Time

The `job-print-after` attribute specifies the calendar date and time after which the job can be scheduled for printing. You may enter values in the format dd:mm:yyyy:HH:MM:SS.

To specify that the file `budget02.txt`print after 5:00 p.m. on March 23, 2002, use the following examples:

```
$ pdpr -x "job-print-after=23:03:2002:17:00:00" budget02.txt
```

If the job has not yet started to print, you can change the time after which it will print. First, use the `pdq` command to obtain the job identifier.

```
$ pdq
job-identifier  job-name     current-job-state intervening-jobs
--------------  --------     ----------------- ----------------
dogear_spl:1517 budget02.txt pending           0
```

Then use the pdmod command to specify a new time (like 7:00 p.m.), using the job identifier as the command operand.

```
$ pdmod -x "job-print-after=02:03:2002:19:00:00" dogear_spl:1517
```

3.3.3 Retaining a Print Job

Print jobs can be retained by the print system after they print or complete with errors. This is useful if you need to print extra copies after a job is complete or if the job does not print correctly. The job-retention-period attribute specifies the period of time following completion of a job that the print system retains the job, its attributes, and its data. By setting this attribute, you can obtain status information after a job has printed and also resubmit the print job.

The job-retention-period attribute specifies a lower boundary on how long the print system retains a job, its attributes, and data. The job-discard-time specifies an upper boundary for retention, regardless of whether the job has printed. Use the format [HH:]MM[:SS] for the time value.

For example, to specify that the job dogear_spl:27 be retained for 60 minutes after printing, use the pdmod command line:

```
$ pdmod -x "job-retention-period=60" dogear_spl:27
```

3.3.4 Automatically Discarding a Print Job

The job-discard-time attribute specifies the calendar date and time of day at which a print job should be discarded, regardless of whether it has printed.

This attribute specifies an upper boundary for retention, regardless of whether the job has printed. The job-retention-period attribute specifies a lower boundary on how long a job, its attributes, and data are retained. Use the format dd:mm:yyyy:HH:MM:SS for the time value.

To specify that the print job for the budget99.txt file should be deleted if it does not print by 5:00 p.m. on February 2, 1999, submit it as follows:

```
% pdpr -x "job-discard-time=02:02:1999:17:00:00" budget99.txt
```

3.4 Specifying Default Attributes for Jobs and Documents

The printing system supports two kinds of objects that supply an initial set of attributes for jobs and documents:

- `initial-value-job` objects, which contain a set of job attributes that are applied to a job.
- `initial-value-document` objects, which contain a set of document attributes that are applied to individual documents in a job.

Initial-value objects make it easy for you to apply commonly-used sets of job and document attributes to your print jobs. Your administrator can create as many initial value objects as necessary. In addition, it is easy to modify the attributes and values they contain.

Initial value objects can be used in two ways:

- Administrators can apply initial-value objects to a logical printer by setting the logical printer's `printer-initial-value-job` and `printer-initial-value-document` attributes. Jobs you submit to the logical printer inherit the attributes contained in the initial-value object.
- You can apply initial value objects to a job or document by specifying them in print requests with the `initial-value-job` and `initial-value-document` attributes.

Only administrators can create or modify initial value objects. If you need to create a a colletion of job and document attributes, refer to Section 3.4.3.

Every site is unique. Check with your administrator for details on your installation's use of such objects.

3.4.1 Setting Default Print Job Attributes

The following example demonstrates how an `initial-value-job` object might be used.

Your administrator uses the `printer-initial-value-job` attribute to apply default job settings to your default logical printer. One of those settings is the `job-sheets=job-copy-start` attribute. Consequently, all jobs sent to that logical printer include job start sheets.

You, however, prefer to have start sheets, end sheets, and document sheets printed with certain jobs. Your administrator created an initial-value-job object called *ivj_full_sheets* for this purpose. It contains the `job-sheets=job-copy-wrap` and `document-sheets=doc-set-start-copy-separate` attributes.

To send the file `quote.txt` to the default printer and apply `ivj_full_sheets` to your print request, enter:

```
$ pdpr -x initial-value-job=ivj_full_sheets quote.txt
```

3.4.2 Setting Default Print Document Attributes

The example below shows how an initial-value-document object might be used.

Your administrator uses a `printer-initial-value-document` attribute to apply document defaults to a logical printer. One of those defaults is the `sides=1` attribute. As a result, all documents submitted to that logical printer are printed one sided.

You, however, occasionally need to print text documents two-sided and two page images per sheet side. To meet this need, your administrator created an initial-value-document object called `ivd_2x2`. It contains the `sides=2` and `number-up=2` attributes.

To send the file `quote.txt` to the logical printer called bulldog and apply initial values in `ivd_2x2` to your print request, enter:

```
$ pdpr -x initial-value-document=ivd_2x2 -p bulldog quote.txt
```

3.4.3 Using an Attribute File

You can predefine specific attributes and initial value objects in a text file and include them with a print command. These attributes files are a convenient way to store and retrieve frequently used combinations of printing attributes and values, making it easier to express complex commands reliably. Refer to the *Advanced Printing Software Command Reference Guide* for more information.

For example, if you need to print five copies of a monthly report for a staff meeting and print them two sided on a printer named pawprint, then create and use the following attributes file, `monthly.attr`:

```
job-comment="wonderful monthly report"
document-format=PostScript
job-copies=5
sides=2
printer-requested=pawprint
```

Use the `-X` option to specify the attributes file. For example, if your report file is mar02.ps, enter this:

```
$ pdpr -X monthly.attr mar02.ps
```

3.5 Requesting Notification of Job Status

You can use the `-N`option with the `pdpr` command to request notification when your job has completed printing. Notification can be via e-mail or a message written to the console window.

Valid values for the `-N` option are:

`email`[:*address*]

`message`[:*address*]

3.5.1 Requesting E-mail Notification

To print file `inventory.txt` on the default logical printer and receive an e-mail notice at your jane.doe@abc.com address after the job has printed, enter:

```
$ pdpr -N email:jane.doe@abc.com inventory.txt
```

The default value for an email address is <your username>@<your DNS domain> .

3.5.2 Requesting Console Message Notification

Specify the `-N` option with the message argument to send a notice to a host's console window, typically the dxconsole console in the CDE, Motif, or X Window environment. Before you can use the message method for notification, you must be running the Console Notification Daemon (`/usr/pd/lib/pdconntf`).

To print the file `cutbacks.txt` and request that a job-completion message be written to the console window on node bulldog, enter:

```
$ pdpr -N message:bulldog cutbacks.txt
```

The default address for console messages is the client hostname.

3.6 Using Preprocessor Filters

Document data frequently needs translation or modification before printing. For example, to print a simple text file on a PostScript printer, the text needs to be translated to PostScript using preprocessor filter programs. The print system supports the following kinds of filter programs:

- Translation filter – a filter that translates one document format to another, typically one supported by a printer. An example of a translation filter is one that translates a file from TIFF to PostScript. The print system automatically invokes a translation filter whenever the document format of your file does not match the native printer language

of the target physical printer. You can also specify a translation filter with the `pdpr` command.

- Modification filter – a filter that modifies the document data stream but does not change the document format. One example is a program that inserts line numbers and page headers in a text file. To use a modification filter, you must specify it with the `pdpr` command.

Your system administrator can set up filters that use common UNIX utilities that read from standard input and write to standard output. The Advanced Printing Software includes a text-to-PostScript translation filter that performs the task of translating text files into the PostScript language. With help from your administrator, you can create or obtain other preprocessor filters.

Your system administrator uses the `filter-definition` attribute to configure preprocessing filters for the print supervisor servers. You can obtain a list of configured filters if you know the name of the supervisor server that controls the physical printers you use.

The following example show how to determine which filters are available for a given logical printer.

- Determine which physical printers are associated with the logical printer in this case, bulldog:

```
$ pdls -c printer -r printer-associated-printer bulldog
printer-associated-printers
---------------------------
bulldog1_pp
bulldog2_pp
```

- Use one of the reported physical printer names to request the name of its associated server, a printer supervisor:

```
$ pdls -c printer -r associated-server bulldog1_pp
associated-server
-----------------
dogear_sup
```

- Request the list of filter definitions available to that printer supervisor:

```
$ pdls -c server -r filter-definition dogear_sup
```

Translation and modification filters are displayed.

You can request that a filter be applied to documents you print by specifying the `modification-filter` attribute for a modification filter, or the `translation-filter` attribute to select a translation filter.

In the following example, the file `c_supprt.c` is printed on the default printer with a modification filter named listing.

```
$ pdpr -x modification-filter=listing c_supprt.c
```

3.6.1 Translating Text Files to PostScript

The text-to-PostScript translation filter (/usr/pd/bin/trn_textps) translates simple-text documents to PostScript. This filter executes whenever you send a simple text document to a printer that supports only the PostScript language.

The text-to-PostScript translation filter provided with Advanced Printing Software offers formatting options, such as number-up printing, page orientation, page length and width, and adjustable margins.

Administrators can set up the `filter-definition` attribute with command option substitutions that relate job and document attributes to translator options. The configuration of the text-to-PostScript translation filter is performed automatically when the administrator uses pd_get_started to setup a supervisor server..

Although translation filters are automatically applied, you can specify particular translation filters if more than one of a given type is configured on your server.

In the following example, the text file `c_supprt.c` is printed on the default PostScript-capable printer with the built-in text-to-PostScript filter. The supervisor filter detects that a translation filter is required to convert the text to PostScript.

```
$ pdpr c_supprt.c
```

In the following example, the file `c_supprt.c` is printed on the default printer with the translation-filter list-to-ps.

```
$ pdpr -x translation-filter=list-to-ps c_supprt.c
```

3.7 Printing to a Specific Physical Printer

When you submit a print job, it is sent to a logical printer that might be associated with several physical printers. The spooler schedules the request to an available physical printer most suited to the needs of the job. You can, however, send a job to a specific physical printer by using the `physical-printers-requested` attribute.

You can determine the names of physical printers supported by a logical printer by using the `pdls` command with the `printer-associated-printers` attribute, as in the following example:

```
$ pdls -c printer -r "printer-associated-printers" printmore
```

This command line returns a list of all physical printers associated with the logical printer printmore. To print your job on one of those physical printers, you must also specify the logical printer, either implicitly with the PDPRINTER environment variable, or explicitly with the `-p` option.

For example, to submit the file `emily.ps` to the physical printer printer1_pp that is associated with logical printer printmore, enter:

```
$ pdpr -p printmore \
-x "physical-printers-requested=printer1_pp" emily.ps
```

4

Managing Your Print Job

This chapter explains how to check the status of, modify, cancel, and resubmit a print job.

4.1 Checking the Status of Your Job

Use the pdq command to request a list of jobs that you have submitted to a logical printer. This information can cover one or all jobs you have submitted and that are currently residing in the queue associated with the printer.

- The pdq command writes the report to standard output.
- If a printer is not specified, pdq lists jobs on the printer named as the PDPRINTER environment variable.

The returned list of jobs is displayed in the order in which the jobs are scheduled to print.

As an end user, you can view only the jobs that you own.

For example, to list all of your jobs in the queue associated with the default logical printer:

```
$ pdq
```

To list all of your jobs in the queue associated with logical printer bulldog, enter:

```
% pdq -p bulldog
```

When you use the pdq command to request status information, the command displays the following job attributes:

- Job identifier (the server name and a number generated by the server)
- Job name
- Current job state
- Number of intervening jobs ahead of yours in the queue
- Logical printer requested
- Physical printer assigned

4.1.1 Job States

Part of the information returned by the pdq command is the job's current state. Table 4–1 describes possible job states.

Table 4–1: Job States

State	Description
completed	The job completed printing or was canceled.
held	The job is being held until the job-hold attribute is set to no.
paused	The job was paused via the pdpause command.
pending	The job is waiting to be scheduled.
printing	The job is printing now.
processing	The job is scheduled for printing and is awaiting a connection to the physical printer.
retained	The job completed printing or failed. The job, its attributes, and data are being retained until a specified period of time elapses. During the retention period, you can resubmit the print job.
terminating	The job has been canceled. The supervisor is terminating its connection to the physical printer.

4.1.2 Displaying a Detailed List of Your Job's Attributes

To display a detailed list of your job's attributes, use the pdq command with the -r verbose option. For example, to display the verbose set of job attributes associated with job 123 on logical printer bulldog, spooler dogear_spl, enter:

```
$ pdq -p bulldog -r verbose dogear_spl:123
```

To display a list of your job's attributes including document attributes, use the pdls command and specify -x scope=1. For example, to display the verbose set of job and document attributes associated with job 123, enter:

```
$ pdls -c job -r verbose -x "scope=1" dogear_spl:123
```

To request all of a job's attributes, including many that are not displayed when you specify verbose, use the -r all option. Use the -s line option to display one attribute per line.

```
% pdls -c job -r all -s line dogear_spl:123
```

4.1.3 Requesting Job Error Information

If your job has not printed correctly you might be able to retrieve additional information by using the `pdls` command to request certain attributes. The job attributes that pertain to job errors are `current-job-state`, `job-state-reasons`, and `job-state-message`.

For example, to request job error information about job 1547 on the default server, enter:

```
$ pdls -c job \
-r "job-id current-job-state job-state-reasons \
job-state-message" 1547
```

If you suspect a problem with the printer device, you may be able to get additional information by requesting certain physical printer attributes.

1. Determine which physical printer was assigned your job.
2. Determine whether the job was assigned to a physical printer, and if so, which one.

   ```
   $ pdls -c job -r printers-assigned 1547
   ```

- If a printer name is displayed, for example, bulldog2_pp request information about that physical printer:

  ```
  $ pdls -c printer -r\
  "printer-state printer-problem-message" bulldog2_pp
  ```

- If no name was returned for the value of the `printers-assigned` attribute, then the job has not yet been scheduled to print.
- If the state is pending, there might be a job or document attribute that cannot be satisfied with the current set of physical printers. You might need to request help from an administrator to check that all required printer attributes are supported and ready. Once an appropriate physical printer is available, your job should print.

4.2 Modifying Your Print Job

Use the `pdmod` command to modify job and document attributes of a job that has not yet started to print.

The following guidelines apply to the `pdmod` command:

- To modify job attributes:
  - Include the job identifier and do not include a document identifier.
  - Specify only job attributes.
- To modify document attributes:

  - Include the document identifier. This is a number less than or equal to the total number of documents in the job.
  - Specify only document attributes.
- To modify job and document attributes:
  - Include the job identifier and document identifier of the specific document you are modifying.
  - Include the specific job and document attributes that are to be changed.

The following examples show pdmod commands being used to modify jobs and documents on the default server.

- To change the copy count to 4 for job 112 on dogear_spl, enter:

  ```
  $ pdmod -n 4 dogear_spl:112
  ```

- To modify the job retention period to 60 minutes for job 113 on the default spooler, enter:

  ```
  % pdmod -x "job-retention-period=60" 113
  ```

- To change the default medium for the first document of job 127 on the default spooler, enter:

  ```
  % pdmod -x "default-medium=a" 127.1
  ```

4.3 Pausing and Resuming Your Job

Use the pdpause command to pause a pending print job or held print job. You cannot pause a job that has started to print. When you pause a job, the job is not submitted to a physical printer for printing.

The following guidelines apply to the pdpause command:

- You can pause only your own jobs.
- You cannot pause a specific document within the job, only the whole job.
- You must specify the job identifier for the job to be paused. If the server is your default server, you need only specify the job number.
- You cannot pause a job after it starts printing.

Some examples of pdpause commands are:

- To pause job 1023, which is waiting to be printed on the default logical printer, enter:

  ```
  $ pdpause 1023
  ```

- To pause job 1153 on dogear_spl, enter:

  ```
  $ pdpause dogear_spl:1153
  ```

Use the pdresume command to resume a job that was paused with the pdpause command. The job then becomes available for scheduling and printing.

The following guidelines apply to the pdresume command:

- You can resume only jobs that were paused with the pdpause command.
- You can resume only your own paused jobs.
- You must specify the job identifier for the job to be resumed. If the server is your default server, you need only specify the job number.

The following examples show the pdresume commad being used:

- To resume job 123, which was submitted to the default printer and then paused, enter:

  ```
  % pdresume 123
  ```

- To resume job 1153 on dogear_spl, enter:

  ```
  % pdresume dogear_spl:1153
  ```

4.4 Resubmitting Your Job

Use the command to requeue a job for printing.

The following guidelines apply to the pdresubmit command:

- You can resubmit only your own jobs.
- The job to be resubmitted must currently be in a pending, held, paused, or retained state.
- You cannot resubmit a job if its state is printing, processing, preprocessing, or completed.
- The new (target) printer must be on the same server as the printer to which the job was originally sent.
- You must specify the job identifier for the job to be resubmitted. If the server is your default server, you need only specify the job number.

The following example shows the pdresubmit command being used:

- To resubmit jobs 2000 and 2001 on the default server to logical printer pawprint, enter:

  ```
  $ pdresubmit pawprint 2000 2001
  ```

4.5 Remove Your Job

Use the pdrm command to remove (cancel) a print job.

The following guidelines apply to the pdrm command:

- You can cancel only your own jobs.
- The removed job is placed in a retained state if the `-r retention_period` option is included in the command line or if the `job-retention-period` attribute has a value greater than zero. Otherwise, the job is placed in a completed state and the document data is deleted.
- If you set a retention period for the job, you can use the `pdresubmit` command to resubmit the job at any time within the specified retention period.
- You must specify the job identifier for the job to be canceled. If the server is your default server, you need only specify the job number.

The following examples show the `pdrm` command being used:

- To cancel and delete job 2000 from the default server, enter:

  ```
  $ pdrm 2000
  ```

- To cancel job *2001* on server *dogear_spl*, and retain the document data for one hour, enter:

  ```
  $ pdrm -r 60 dogear_spl:2001
  ```

A

Attribute Reference

The print system provides attributes for changing the characteristics of each print job. This appendix provides summaries of several commonly used attributes organized into the following categories:

- Job and document attributes determine the printer requirements of jobs and documents.
- Text job attributes provide controls for printing simple text jobs.
- Physical printer attributes define the capabilities of the printer device the object represents.

For a detailed directory of all attributes and their associated values, refer to the *Advanced Printing Software Command Reference Guide*.

A.1 Common Job and Document Attributes

The following job and document attributes are available for use with print commands. The attribute always applies to the file name and filenames entered on the command-line after the attribute. See the accompanying examples.

`additional-production-instructions`

Specifies lpd processing options that are not representable as attributes.

Intended for use only by inbound and outbound gateways.

`copy-count`

Requests a number of copies of the document. For example:

```
pdpr -x "copy-count=2" report.txt
```

`default-input-tray`

Requests that the document is to be printed on media drawn from the named input tray. Input tray names are defined in the printer's `input-tray-supported` attribute. For example:

```
pdpr -x "default-input-tray=bottom" report.txt
```

`default-medium`

Requests that the document is to be printed on the named media. Media names are defined in the printer's `media-supported` attribute. For example:

```
pdpr -x "default-medium=iso-a4-white" report.txt
```

`document-sheets`

Specifies whether to print auxiliary sheets at the beginning of each document in a job. If you specify `doc-set-start-copies-separate`, a separator sheet precedes each copy of the document. For example:

```
pdpr -x "document-sheets=doc-set-start-copies-separate" report.txt
```

`job-comment`

Specifies a text comment for a print job. If you use the `job-sheets` attribute to select job start sheets, then the `job-comment` is printed on them. For example:

```
pdpr -x "job-sheets=job-copy-start \
job-comment='final draft of secret report'" report.txt
```

`job-copies`

Requests a number of collated copies of all documents in the entire job. For example:

```
pdpr -x "job-copies=3" report.txt forecast.txt budget.txt
```

`job-name`

Supplies a name for a print job. The job name is printed on job start sheets and used in notification and logging messages. If you do not specify a `job-name`, it defaults to the file name in a one document job or to the name of the first file in a multi-document job. For example:

```
pdpr -x "job-name='copy for Pokey'" report.txt
```

`job-print-after`

Specifies the calendar date and time after which the job should be scheduled. Use the format dd:mm:yyyy:HH:MM:SS. When the specified date and time arrive, the job is scheduled for printing. For example:

```
pdpr -x "job-print-after=31:12:1999:23:59" report.txt
```

`job-priority`

Specifies a print job scheduling priority value. Jobs with higher priorities are scheduled to print before jobs with lower priorities. For example:

```
pdpr -x job-priority=20 new_job.txt
```

`job-retention-period`

Specifies the period of time following job completion that the system retains the job, its attributes, and data. Use the format [HH:]MM[:SS]. By setting this attribute, you can obtain status information after your job has printed. It also allows you to print the job again, possibly with modified attributes. For example:

```
pdpr -x "job-retention-period=01:00" report.txt
```

`printer-setup-module`

Specifies one or more files used to set up printer modes or functions prior to printing a document.

`job-sheets`

Specifies the auxiliary sheets that will print with a job. If you specify `job-copy-start`, a start sheet prints in front of every copy of the job. If you specify `job-copy-wrap`, a start and end sheet prints for every copy of the job. For example:

```
pdpr -x "job-sheets=job-copy-start" report.txt
```

`number-up`

Requests that the document is to be printed with multiple page images on one side of the sheet. This is done by reducing the size of the printed page. For example, `number-up=2` reduces two pages so they print side by side on one sheet. Values are defined in the printer's `numbers-up-supported` attribute. A value of 0 is equivalent to no `number-up` processing. This feature is generally limited to the printing of text files on PostScript printers. For example:

```
pdpr -x "number-up=4" report.txt
```

`output-bin`

Requests that the job be deposited in the printer's specified output bin. Output bin names are defined in the printer's `output-bins-supported` attribute. For example:

```
pdpr -x "output-bin=side" report.txt
```

`sides`

Requests that the document is to be printed one-sided or two-sided. Values of 1 or 2 or both are defined in the printer's `sides-supported` attribute. For example:

```
pdpr -x "sides=2" report.txt
```

A.2 Text Document Attributes

The print system supports several attributes that are used primarily for documents that are free of formatting instructions, that is they contain only text. These attributes provide some control over the appearance and placement of text in the printed document. Some attributes are specific to document formats such as PCL or ESC/P and to printers that use those formats. Others apply only when using the text-to-PostScript translation filter supplied and Advanced Printing Software with a PostScript-capable printer.

For the following attributes, only *content-orientation* applies to both formats. See the accompanying examples.

`bottom-margin`

Specifies the distance in lines between the bottom edge of the logical page and the bottom edge of the text area when held in the intended reading orientation. For example:

```
pdpr -x "bottom-margin=3" report.txt
```

`content-orientation`

Specifies the most significant orientation of the document. Choices include:
portrait
landscape
reverse-portrait
reverse-landscape

For example:

```
pdpr -x "content-orientation=landscape" report.txt
```

`header-text`

Specifies the text that is to be printed on the first line of each printed page. The `header-text` could be the title of the document. For example:

```
pdpr -x "header-text='Favorite Pumpkin Recipes'" report.txt
```

`left-margin`

Specifies the distance as the number of characters between the left edge of the logical page and the left edge of the text area when held in the intended reading position. For example:

```
pdpr -x "left-margin=8" report.txt
```

`length`

Specifies the page length of the text area as a number of lines. For example:

```
pdpr -x "length=60" report.txt
```

`number-pages`

Indicates whether to print page numbers on the document pages. The value may be yes or no. For example:

```
pdpr -x "number-pages=yes header-text='Final Draft'" report.txt
```

`number-up`

Requests that the document is to be printed with multiple page images on one side of the sheet. This is done by reducing the size of the printed page. For example, number-up=2 reduces two pages so they print side by side on one sheet. Choices include 0, 1, 2, and 4. The value 0 is equivalent to no `number-up` processing. For example:

```
pdpr -x "number-up=4" report.txt
```

`right-margin`

Specifies the distance as the number of characters between the right edge of the logical page and the right edge of the text area when held in the intended reading position. For example:

```
pdpr -x "left-margin=6 right-margin=6" report.txt
```

`top-margin`

Specifies the distance in lines between the top edge of the logical page and the top edge of the text area when held in the intended reading orientation. For example:

```
pdpr -x "top-margin=5 bottom-margin=6" report.txt
```

`width`

Specifies the width of the text area as the number of characters. This is the maximum line width before text wrapping occurs. For example:

```
pdpr -x "length=56 width=132 content-orientation=landscape" report.txt
```

A.3 Commonly Used Logical Printer Attributes

Logical printers reside in and are controlled by spooler processes. As an end user, you normally cannot change logical printer attributes, but you can query them. Use the `pdls` command and specify the logical printer name as the command operand.

When you use the `pdls` command to display properties of a logical printer, you can request attributes associated with the printer. For example, requesting the value of the `printer-state` attribute lets you know if the printer is idle, printing, needs attention, or is in some other state.

You can use the following logical printer attributes with the `pdls` command. For example, the following command requests a list of the input trays that are supported for a logical printer named printer_1:

```
$ pdls -c printer -r "input-trays-supported" printer_1
```

`availability`

Indicates the general availability of an object. It is set to none if the object is disabled and normal if the object is enabled.

`character-sets-supported`

Identifies the character set encodings supported by the printer.

On job submission, the spooler checks the character set specified for a document against the logical printer's `character-sets-supported` attribute. If there is no match, the spooler rejects the print request. Refer to the *Advanced Printing Software Command Reference Guide* for a list of values.

`content-orientations-supported`

Specifies the document content orientations supported by the printer. This attribute's values must include any content orientation for a document directed to the printer. Valid values are portrait, landscape, reverse-portrait, and reverse-landscape.

`document-formats-supported`

Specifies the document formats supported by the printer. This attribute must contain a value corresponding to the value of the `document-format` attribute of a document submitted to this printer. Refer to the *Advanced Printing Software Command Reference Guide* for a list of values.

`document-sheets-supported`

Specifies the auxiliary sheets supported by this printer.This attribute must contain a value corresponding to the value of the `document-sheets` attribute of a job submitted to this printer. Valid values are none and doc-set-start-copies-separate.

`enabled`

Indicates whether the specified object is enabled to accept print requests (value=yes). This attribute is set with `pdenable` or `pddisable`. When an object is created, it is disabled by default (value=no).

For a server to accept print requests, the server's, the associated queue's, and the specified printer's enabled attributes must be set to yes.

`finishings-supported`

Identifies the per-document finishings supported on the printer. This attribute must contain a value corresponding to the value of the `finishing` attribute of a document submitted to this printer.

Refer to the *Advanced Printing Software Command Reference Guide* for a list of values.

`fonts-supported`

Identifies the font resources supported by the printer.

On job submission, the spooler checks the font specified for a document against the logical printer's `fonts-supported` attribute. If there is no match, the spooler rejects the print request.

`highlight-colours-supported`

Indicates the values of highlight colors supported on this printer.

This attribute must contain a value corresponding to the value of the `highlight-colour` attribute of a document submitted to this printer. Valid values are red, blue, green, cyan, magenta, yellow, cardinal, royalblue, ruby, violet, black, or name.

`input-trays-supported`

Identifies the input trays supported on this printer.

This attribute must contain a value corresponding to the value of the `default-input-tray` attribute of a document submitted to this printer. Valid values are top, middle, bottom, envelope, manual, large-capacity, main, side, or 1, 2, 3, and 4.

`job-finishings-supported`

Identifies the job-level finishing supported by this printer.

On job submission, the spooler checks the finishing specified for a job against the logical printer's `job-finishings-supported` attribute.

If there is no match, the spooler rejects the print request. Refer to the *Advanced Printing Software Command Reference Guide* for a list of valid values.

`job-sheets-supported`

Specifies the auxiliary sheets supported by this printer. This attribute must contain a value corresponding to the value of the `job-sheets` attribute of a job submitted to this printer. Values are none, job-copy-start, and job-copy-wrap.

`jobs-pending`

Specifies the number of outstanding jobs in the queue. Outstanding jobs are jobs with a `current-job-state` value of pending, held, or paused.

`maximum-copies-supported`

Indicates the maximum number of copies of a document that can be printed on this printer. This includes document copies specified via the attributes *copy-count* and *job-copies*. A value of zero (0) or empty indicates no limit.

`media-supported`

Identifies the media supported by the printer.

On job submission, the spooler checks the medium specified for a document against the logical printer's `media-supported` attribute. If there is no match, the spooler rejects the print request. Refer to the *Advanced Printing Software Command Reference Guide* for a list of valid values.

`message`

Supplies a human–readable string intended to indicate to users something about an object's state. This attribute can be used to indicate to users why an object is unavailable or when it is expected to be ready.

The `-m` option can also be used to attach a human readable message to a job. Users can retrieve the message with the `pdls` command.

`numbers-up-supported`

Indicates valid values for the document attribute `number-up`. Values are 0, 1, 2, and 4.

`output-bins-supported`

Identifies the output bins supported on this printer. The value of this attribute can be an OID, a name, or a number.

This attribute must contain a value corresponding to the value of the `output-bin` attribute of a job submitted to this printer. Refer to the *Advanced Printing Software Command Reference Guide* for a list of values.

`page-select-supported`

Indicates the types of page identifiers supported by this printer. Numeric or alphanumeric page identifiers are used to specify one or more sequences of pages to be printed.

The values for `page-select-supported` must include the value of `page-select` used by a document submitted to this printer.

`print-colour-types-supported`

Identifies the colors that are supported on this printer. Values are black-and-white, highlight-colour, and full color.

On job submission, the spooler checks the print color types specified for a job against the logical printer's `print-colour-types-supported` attribute. If there is no match, the spooler rejects the print request.

`printer-associated-printers`

Identifies the physical printers associated with this logical printer. This attribute is updated when the printer's `associated-queue` attribute is modified. It is checked for end-to-end consistency when the printer is enabled.

`printer-initial-value-document`

Identifies an initial-value-document on the server for use on this logical printer. This attribute is used if the document does not specify an `initial-value-document` object.

`printer-initial-value-job`

Identifies an initial-value-job on the server for use on this logical printer. The attribute is used if the job does not specify an `initial-value-job` object.

`printer-locations`

Identifies the location of the printer.

`printer-name`

Specifies a unique name for a printer.

`printer-problem-message`

Some printers produce a text message describing a problem. In these cases, the supervisor places the message in the `printer-problem-message` attribute.

`printer-realization`

Identifies if the printer is logical or physical. A printer created on a spooler is logical. A printer created on a supervisor is physical.

`printer-state`

Identifies the current state of the printer. Values are unknown, idle, printing, needs-attention, paused, shutdown, timed-out, connecting-to-printer, and needs-key-operator.

`printers-ready`

Identifies the physical printers ready to be used with the specified logical printer.

`sides-supported`

Indicates the values of sides supported by this printer. This attribute must contain a value (1 or 2) corresponding to the value of the `sides` attribute of a document submitted to the specified printer.

B

BSD Print System Command Equivalents

This appendix lists common BSD print system requests and their equivalent Advanced Printing Software command.

Table B–1: BSD and Advanced Printing Software Command Equivalents

BSD Command and Option, and Value	Advanced Printing Software Command, Option, and Value	Action
`lpr -P`*`printer_name`*	`pdpr -p` *`printer_name`*	Submits the job to the specified logical printer.
`lpr -#`*`number`*	`pdpr -n` *`copies`*	Specifies the number of job copies you want printed.
`lpr -m`	`pdpr -N` email	Delivers messages regarding the job by electronic mail.
`lpr -J`*`job`*	`pdpr -t` *`job_name`*	Specifies a name for the job. The name prints on the banner page of the output. If this option is omitted, the name of the first file is used for the job name.
`lpr -h`	`pdpr -x` "job-sheets=none"	Suppresses printing of a banner page (job start sheet). Some administrators might configure their printers to require printing of job start sheets.
`lprm` *`request_id`*	`pdrm` *`job_id`*	Cancels the job specified by *`job_id`*.
`lpq -P` *`printer_name`*	`pdq -p` *`printer_name`*	Lists the status of print jobs for the specified printer.
`lpq -l`	`pdq -r` all `-s` line	Lists information in long format, including the name of the host from which the print request originated.

Index

Q

R

S

T

U

V