

Advanced Printing Software

User Guide

Part Number: AA-PXUSR-GD

January 1999

Product Version: Advanced Printing Software Version
1.0

Operating System and Version: DIGITAL UNIX, Version 4.0D or later

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

© Digital Equipment Corporation 1999
All rights reserved.

The following are trademarks of Compaq Computer Corporation: ALL-IN-1, Alpha AXP, AlphaGeneration, AlphaServer, AltaVista, ATMworks, AXP, Bookreader, CDA, DDIS, DEC, DEC Ada, DEC Fortran, DEC FUSE, DECnet, DECstation, DECsystem, DECterm, DECUS, DECwindows, DTIF, Massbus, MicroVAX, OpenVMS, POLYCENTER, PrintServer, Q-bus, StorageWorks, TruCluster, ULTRIX, ULTRIX Mail Connection, ULTRIX Worksystem Software, UNIBUS, VAX, VAXstation, VMS, XUI, and the Compaq logo.

PostScript is a trademark of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

PrintXchange is a trademark of Xerox Corporation.

Contents

About this Guide

1 Introduction

1.1	Benefits of Using Advanced Printing Software	1-1
1.2	Advanced Printing Software Terminology	1-1
1.3	Access to the User Interfaces	1-2
1.4	Command Syntax and Elements	1-3
1.5	Specifying Options and Arguments	1-3
1.6	Operands and Classes	1-4
1.7	Attributes	1-4
1.7.1	Default Values for Attributes	1-5
1.7.2	Syntax for Attribute Values	1-5
1.7.2.1	Single Value Attributes	1-5
1.7.2.2	Multi-value Attributes	1-5
1.7.2.3	Complex Attributes	1-5
1.7.3	Abbreviating Attributes and Values	1-6
1.8	Job and Document Identifiers	1-6
1.9	Getting Help for Commands	1-7

2 Determining Printer Resources

2.1	Defining Your Default Logical Printer	2-1
2.2	Accessing the Default Logical Printer	2-2
2.3	Accessing a Specific Logical Printer	2-2
2.4	Determining Properties of Printers	2-3
2.4.1	Requesting Specific Attributes with the <code>pdls</code> Command ..	2-4
2.5	Logical Printer Defaults	2-4
2.5.1	Using <code>printer-initial-value-job</code> Objects	2-5
2.5.2	Using <code>printer-initial-value-document</code> Objects	2-5
2.5.3	Checking a Logical Printer for Initial Value Objects	2-5
2.5.4	Using an Attributes File	2-6

3 Printing Your Job

3.1	Submitting a Print Job	3-1
-----	------------------------------	-----

3.2	Specifying Attributes for a Job	3-1
3.2.1	Specifying the Number of Copies	3-2
3.2.2	Specifying Two-sided Printing	3-2
3.2.3	Specifying the Document Format	3-3
3.2.4	Specifying Media	3-3
3.2.5	Specifying Input Trays	3-4
3.2.6	Specifying Output Bins	3-4
3.2.7	Specifying Finishing Processes	3-5
3.2.8	Printing from Standard Input	3-5
3.3	Specifying Job Scheduling Attributes	3-5
3.3.1	Holding a Print Job	3-5
3.3.2	Printing a Job After a Certain Time	3-6
3.3.3	Retaining a Job	3-6
3.3.4	Automatic Discarding of Jobs	3-7
3.4	Specifying Defaults for Jobs and Documents	3-7
3.4.1	Using initial-value-job Objects	3-8
3.4.2	Using initial-value-document Objects	3-8
3.4.3	Using an Attributes File	3-9
3.5	Requesting Notification of Job Status	3-9
3.5.1	Requesting Notification by E-mail	3-9
3.5.2	Requesting Notification by Console Message	3-9
3.6	Using Preprocessor Filters	3-10
3.6.1	Text-to-PostScript Translation Filter	3-11
3.7	Printing to a Specific Physical Printer	3-12

4 Managing your Print Job

4.1	Checking the Status of Your Job	4-1
4.1.1	Job States	4-1
4.1.2	Obtaining a Detailed List of Your Job's Attributes	4-2
4.1.3	Checking for Job Errors	4-2
4.2	Modifying Your Job	4-3
4.3	Pausing and Resuming Your Job	4-4
4.4	Resubmitting Your Job	4-5
4.5	Canceling Your Job	4-6

A Attribute Reference

A.1	Common Job and Document Attributes	A-1
A.2	Attributes for Simple Text Jobs	A-3

A.3	Commonly Used Logical Printer Attributes	A-5
-----	------------------------------------------------	-----

B BSD Print System Command Equivalents

Index

Examples

1-1	Command Line Syntax	1-3
-----	---------------------------	-----

Tables

1-1	End User Commands	1-4
4-1	Job States	4-2
B-1	BSD and Advanced Printing Software Command Equivalents	B-1

About this Guide

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

Audience

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

Related Software

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

Organization

This manual consists of the following sections:

- Chapter 1 – an introduction to the manual and an overview of the Advanced Printing Software
- Chapter 2 – information on determining printer resources
- 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 guide, 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.

Conventions

This document 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.
<i>file</i>	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, <code>cat(1)</code> indicates that you can find information on the <code>cat</code> command in Section 1 of the reference pages.

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 doesn't 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 legacy BSD-based printing system. Advanced Printing Software accepts print requests issued by lpr/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 will help you understand 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 HPLaserJetIII or 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 that is described in this book, the Advanced Printing Software contains a graphical user interface that can be used to submit print jobs and monitor print jobs.

pdprint is the GUI program used to submit print jobs and pdprintinfo is the GUI program used to obtain 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:

- pdprint can be accessed by typing 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.

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

1.4 Command Syntax and Elements

The syntax for all print system commands has the format shown in Example 1-1.

Example 1-1: Command Line Syntax

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

The four syntax elements are:

- Command name
- Option
- Option argument
- Object instance

The complete command looks like this: `pdpr -n 2 opus1.txt`

1.5 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 spaces.
- Commands interpret options and option arguments in the order in which they appear on the command line.
- Not all options work with all commands. Be sure to look at the detailed description of each command to determine which options are valid for that command.

1.6 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 of the commands that take operands can perform an operation on different *classes* of print system object. 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
<code>pdls</code>	Lists print object attributes.
<code>pdmod</code>	Modifies previously submitted print jobs or documents.
<code>pdpause</code>	Pauses your own print jobs.
<code>pdpr</code>	Submits a print job.
<code>pdq</code>	Reports on or obtains status of print jobs.
<code>pdresubmit</code>	Resubmits print jobs to another logical printer.
<code>pdresume</code>	Resumes your own print jobs.
<code>pdrm</code>	Removes (that is, cancels) your own print jobs.

From this table, you can infer, for example, that `pdls` and `pdls -c job` produce the same results.

1.7 Attributes

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.

1.7.1 Default Values for Attributes

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 clear an attribute of all values, use the `pdmod` or `pdset` command and specify the attribute name followed by `={}`.
- 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*.

1.7.2 Syntax for Attribute Values

Some attributes can have only one value at a time (single-valued). Others can have multiple values (multi-valued). 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, multi-valued, and complex attributes.

1.7.2.1 Single Value Attributes

The syntax for single valued attributes is:

```
attribute=value
```

For example:

```
copy-count=2
```

1.7.2.2 Multi-value Attributes

The syntax for multi-valued attributes is:

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

For example:

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

1.7.2.3 Complex Attributes

The syntax for complex attributes is:

```
attribute="{attribute=value attribute=value  
attribute=value}"
```

For example:

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

1.7.3 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 such 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.

1.8 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 multi-document job.

Some commands require you to identify a specific document within a multi-document job. A document identifier is expressed as a job identifier, a period, and a document number. Within each multi-document 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*.

1.9 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

Determining Printer Resources

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 of 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 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.
Since 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, 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, then save the file:

```
% setenv PDPRINTER PrinterName
```

2. Apply the changes to the `.login` file by entering this 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, then 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 will be 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 Accessing the 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 request such a list, enter:

```
% pdq
```

2.3 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 of 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
```

2.4 Determining Properties of Printers

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 is capable of accepting 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 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
```

2.4.1 Requesting Specific Attributes with the pdls Command

You can request 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
```

2.5 Logical Printer Defaults

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 collection 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 at a particular logical printer.

An *initial-value-document* object is a similar collection 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.

2.5.1 Using printer-initial-value-job Objects

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, include job start sheets.

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

2.5.2 Using printer-initial-value-document Objects

Similarly, 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.

You can also specify `initial-value-document` objects, such as a different number of copies, when you submit a print job.

2.5.3 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 will list attributes that are defined by the initial value objects. For example, the command to list initial-value-job information might produce the following output:

```
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
```

2.5.4 Using an Attributes File

You can predefine specific attributes in text files called attribute files. You can then include them in a printer command by using the `-X` option to specify an attributes 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 one, double-sided, and without document sheets:

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

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 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 submits *report.txt* and *table.txt* to the printer named *pawprint*.

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

3.2 Specifying Attributes for a Job

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.

There are several ways to specify more than one attribute in a print request. You can bundle them together inside quotes, following the `-x` option; you can specify each attribute with its own `-x` option; or you can store the attributes in an attribute file and specify that file with the `-X` (capital 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 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 copies` 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 specify a different number of copies of documents in a job, use the `copy-count` attribute. The following example prints two copies of *slides.ps* and three copies of *handout.ps*.

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

3.2.2 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:

```
% 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 -x sides=2 budget99.txt
```

The print job goes to the printer that can print two-sided copies. If the printer doesn't have a value for *sides-supported*, then you cannot use the *sides* attribute for your job. The printer prints the document according to control instructions inside the document, 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.

3.2.3 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
```

3.2.4 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 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.

3.2.5 Specifying Input Trays

Another way of selecting printing media is to use `pdpr` 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.

3.2.6 Specifying Output Bins

You can specify the output bin for your print job by using `pdpr` 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
```

3.2.7 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:

```
% 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
```

3.2.8 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:

```
% 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 Certain 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 a job print after 5:00 pm on March 2, 1999, use the following examples:

```
% pdpr -x "job-print-after=02:03:1999:17:00:00" budget99.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 budget99.txt pending           0
```

Then use the `pdmod` command to specify a new time (like 7 PM), using the job identifier as the command operand.

```
% pdmod -x "job-print-after=02:03:1999:19:00:00" dogear_spl:1517
```

3.3.3 Retaining a 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 like this:

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

3.3.4 Automatic Discarding of Jobs

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 *budget99.txt* 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=28:02:1999:17:00:00" budget99.txt
```

3.4 Specifying Defaults 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 all documents in 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.

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

3.4.1 Using initial-value-job Objects

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 Using initial-value-document Objects

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 document called *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 Attributes File

You can predefine specific attributes and initial value objects in a text file and include them in a print command. These attribute files serve as 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 attribute 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 *mar98.ps*, enter this:

```
% pdpr -X monthly.attr mar98.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 Notification by E-mail

To print a file named *inventory.txt* on the default logical printer and receive an e-mail notice at your *end-user@paper-jam.com* address after the job has printed, use the following example:

```
% pdpr -N email:end-user@paper-jam.com inventory.txt
```

3.5.2 Requesting Notification by Console Message

Specifying `-N` with the message argument sends notices to a host's console window, typically `dxconsole` 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`), and you must give the server host permission to write messages to the notified host's X server. Use the `xhost` command to determine whether the print system server host is listed among those in the `xhost` list. If it is not, you can add it with the `xhost +nodename` command.

For example, if your print system server is running on host *dogear*, you would enter the following command to allow messages to be sent to your console:

```
% xhost +dogear
```

You can verify that the print server host is added by again entering the `xhost` command without any arguments.

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

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

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.
- *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 in 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 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 examples show how to determine which filters are available for a given logical printer. First, you need to determine which physical printers are associated with the logical printer.

```
% 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
```

Now you can request the list of filter definitions available to that printer supervisor:

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

All filters, translation and modification, 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 Text-to-PostScript Translation Filter

The text-to-PostScript translation filter 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.

Administrators can set up the *filter-definition* attribute with command option substitutions that relate print system attributes to translator options. The configuration of the text-to-PostScript translation filter is performed automatically when the server software is set up for the first time.

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 next example, the file *c_supprt.c* is printed to the default printer with translation-filter *list-to-ps* performing the translation from text to PostScript.

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

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

3.7 Printing to a Specific Physical Printer

When you submit a print job, it is sent to a logical printer that may 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 a *pdls* command with the *printer-associated-printers* attribute, as in the following example:

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

That command returns a list of all physical printers that are 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*, use the following command:

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

Managing your Print Job

4.1 Checking the Status of Your Job

Use the `pdq` command to request a list of jobs that you submitted to a logical printer. This information can cover one or all jobs you have submitted and currently residing in the queue associated with the printer. The returned list of jobs is displayed in the order in which the jobs are scheduled to print.

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

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, use the `pdq` command:

```
% pdq
```

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

```
% 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. The following table describes possible job states.

Table 4–1: Job States

State	Description
<i>completed</i>	The job completed printing or was canceled.
<i>held</i>	The job is being held until the <i>job-hold</i> attribute is set to <i>no</i> .
<i>paused</i>	The job was paused via the <i>pdpause</i> command.
<i>pending</i>	The job is waiting to be scheduled.
<i>printing</i>	The job is printing now.
<i>processing</i>	The job is scheduled for printing and is awaiting a connection to the physical printer.
<i>retained</i>	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 may resubmit the print job.
<i>terminating</i>	The job has been canceled. The supervisor is terminating its connection to the physical printer.

4.1.2 Obtaining 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
```

If you want 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 the 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 instead of wrapping to the width of your window, which is harder to read.

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

4.1.3 Checking for Job Errors

If your job has not printed correctly, for any reason, you may 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, you can enter a command like the following:

```
% 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. First, determine which physical printer was assigned your job. Using the above example, the following command determines 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*, use a `pdls` command like the following to 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 may need to request help from an administrator, and 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 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, do the following:
 - Include the job identifier and do not include a document identifier.
 - Specify only job attributes in the command.
- To modify document attributes, do the following:
 - 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 in the command.

- To modify job and document attributes, do the following:
 - 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 or held print job. You cannot pause a job that has already 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 reschedule the paused job until you resume it via the `pdresume` command.
- 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 use this command only on jobs that were paused via 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.

Some examples of `pdresume` commands are:

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

```
% pdresume 123
```

To resume job *1153* on *dogear_spl*, use:

```
% pdresume dogear_spl:1153
```

4.4 Resubmitting Your Job

Use the `pdresubmit` command to resend a job to another printer on the same spooler.

The following guidelines apply to the `pdpause` 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.

An example of the `pdresubmit` command is:

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

```
% pdresubmit pawprint 2000 2001
```

4.5 Canceling 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 canceled 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.

Examples of the `pdrm` command follow:

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(s) entered on the command line after the attribute. See the accompanying examples.

content-orientation

Requests that the document is to be printed in *landscape* or *portrait* orientation. These values are defined in the printer's *content-orientations-supported* attribute and apply to simple text documents only. For example:

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

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 Dave'" 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=12:31:1999:23:59" report.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
```

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 Attributes for Simple Text Jobs

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 uses those formats. Others apply only when using the text-to-PostScript translation filter supplied with 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. The distance may be a decimal number like 3.14 or 6.28. For example:

```
pdpr -x "bottom-margin=1.5" report.txt
```

content-orientation

Specifies the most significant orientation of the document. Choices include the following:

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. The distance may be a decimal number like 1.2 or 2.5. 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 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 <i>none</i> if the object is disabled and <i>normal</i> if the object is enabled.
character-sets-supported	<p>The <i>character-sets-supported</i> attribute identifies the character set encodings supported by the printer.</p> <p>On job submission, the spooler checks the character set specified for a document against the logical printer's <i>character-sets-supported</i> attribute. If there is no match, the spooler rejects the print request. Refer to the <i>Advanced Printing Software Command Reference Guide</i> for a list of valid values.</p>
content-orientations-supported	The <i>content-orientations-supported</i> attribute 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	The <i>document-formats-supported</i> attribute specifies the document formats supported by the printer. This attribute must contain a value corresponding to the value of the <i>document-format</i> attribute of a document submitted to this printer. Refer to the <i>Advanced Printing Software Command Reference Guide</i> for a list of valid values.
document-sheets-supported	<p>The <i>document-sheets-supported</i> attribute specifies the auxiliary sheets supported by this printer.</p> <p>This attribute must contain a value corresponding to the value of the <i>document-sheets</i> attribute of a job submitted to this printer. Valid values are <i>none</i> and <i>doc-set-start-copies-separate</i>.</p>

enabled	<p>The <i>enabled</i> attribute indicates whether the specified object is enabled to accept print requests (value=yes). This attribute is set with <i>pdenable/pddisable</i>. When an object is created, it is disabled by default (value=no).</p> <p>In order for a server to accept print requests, both the server's and the specified printer's <i>enabled</i> attributes must be set to yes.</p>
finishings-supported	<p>The <i>finishings-supported</i> attribute identifies the per-document finishings supported on the printer. This attribute must contain a value corresponding to the value of the <i>finishing</i> attribute of a document submitted to this printer.</p> <p>Refer to the <i>Advanced Printing Software Command Reference Guide</i> for a list of valid values.</p>
fonts-supported	<p>The <i>fonts-supported</i> attribute identifies the font resources supported by the printer.</p> <p>On job submission, the spooler checks the font specified for a document against the logical printer's <i>fonts-supported</i> attribute. If there is no match, the spooler rejects the print request.</p>
highlight-colours-supported	<p>The <i>highlight-colours-supported</i> attribute indicates the values of highlight colors supported on this printer.</p> <p>This attribute must contain a value corresponding to the value of the <i>highlight-colour</i> attribute of a document submitted to this printer. Valid values are red, blue, green, cyan, magenta, yellow, cardinal, royalblue, ruby, violet, black, or name.</p>
input-trays-supported	<p>The <i>input-trays-supported</i> attribute identifies the input trays supported on this printer.</p> <p>This attribute must contain a value corresponding to the value of the <i>default-input-tray</i> attribute of a document submitted to this printer. Valid values are top, middle, bottom, envelope, manual, large-capacity, main, side, 1, 2, 3, 4.</p>

job-finishings-supported	The <i>job-finishings-supported</i> attribute 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 <i>job-finishings-supported</i> attribute. If there is no match, the spooler rejects the print request. Refer to the <i>Advanced Printing Software Command Reference Guide</i> for a list of valid values.
job-sheets-supported	The <i>job-sheets-supported</i> attribute specifies the auxiliary sheets supported by this printer. This attribute must contain a value corresponding to the value of the <i>job-sheets</i> attribute of a job submitted to this printer. Valid values are none, job-copy-start, and job-copy-wrap.
jobs-pending	The <i>jobs-pending</i> attribute specifies the number of outstanding jobs in the queue. Outstanding jobs are jobs with a <i>current-job-state</i> value of pending, held, or paused.
maximum-copies-supported	The <i>maximum-copies-supported</i> attribute indicates the maximum number of copies of a document that can be printed on this printer. This includes document copies specified via the attributes <i>copy-count</i> and <i>job-copies</i> . A value of zero or empty indicates no limit.
media-supported	The <i>media-supported</i> attribute identifies the media supported by the printer. On job submission, the spooler checks the medium specified for a document against the logical printer's <i>media-supported</i> attribute. If there is no match, the spooler rejects the print request. Refer to the <i>Advanced Printing Software Command Reference Guide</i> for a list of valid values.
message	The <i>message</i> attribute supplies a human readable string intended to indicate to users something about an object's state. This attribute may be used to indicate to users why an object is unavailable or when it is expected to be ready.

	<p>The <code>-m</code> option can also be used to attach a human readable message to a job. Users can retrieve the message with the <code>pdls</code> command.</p>
numbers-up-supported	<p>The <i>numbers-up-supported</i> attribute indicates valid values for the document attribute <i>number-up</i>. Valid values are 0, 1, 2, and 4.</p>
output-bins-supported	<p>The <i>output-bins-supported</i> attribute identifies the output bins supported on this printer. The value of this attribute may be an OID, a name, or a number.</p> <p>The <i>output-bins-supported</i> attribute must contain a value corresponding to the value of the <i>output-bins</i> attribute of a job submitted to this printer. Refer to the <i>Advanced Printing Software Command Reference Guide</i> for a list of valid values.</p>
page-select-supported	<p>The <i>page-select-supported</i> attribute 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.</p> <p>The values for <i>page-select-supported</i> must include the value of <i>page-select</i> used by a document submitted to this printer.</p>
print-colour-types-supported	<p>The <i>print-colour-types-supported</i> attribute identifies the colors that are supported on this printer. Valid values are lack-and-white, highlight-colour, and full color.</p> <p>On job submission, the spooler checks the print color types specified for a job against the logical printer's <i>print-colour-types-supported</i> attribute. If there is no match, the spooler rejects the print request.</p>
printer-associated-printers	<p>The <i>printer-associated-printer</i> attribute identifies the physical printers associated with this logical printer. This attribute is updated when the printer's <i>associated-queue</i> attribute is modified. It is checked for end-to-end consistency when the printer is enabled.</p>

printer-initial-value-document	<p>The <i>printer-initial-value-document</i> attribute identifies an initial-value-document in the server for use on this logical printer.</p> <p>This attribute is used if the document does not specify an <i>initial-value-document</i> object.</p>
printer-initial-value-job	<p>The <i>printer-initial-value-job</i> attribute identifies an initial-value-job in the server for use on this logical printer.</p> <p>The <i>printer-initial-value-job</i> is used if the job does not specify an <i>initial-value-job</i> object.</p>
printer-locations	<p>The <i>printer-locations</i> attribute identifies the location of the printer.</p>
printer-name	<p>The <i>printer-name</i> attribute specifies a unique name for a printer.</p>
printer-problem-message	<p>Some printers produce a text message describing a problem. In these cases, the supervisor places the message in the <i>printer-problem-message</i> attribute.</p>
printer-realization	<p>The <i>printer-realization</i> attribute identifies if the printer is logical or physical. A printer created on a spooler is logical. A printer created on a supervisor is physical.</p>
printer-state	<p>The <i>printer-state</i> attribute identifies the current state of the printer. Valid values are unknown, idle, printing, needs-attention, paused, shutdown, timed-out, connecting-to-printer, and saturated.</p>
printers-ready	<p>The <i>printers-ready</i> attribute identifies the physical printers ready to be used with this logical printer.</p>
sides-supported	<p>The <i>sides-supported</i> attribute indicates the values of sides supported by this printer. This attribute must contain a value (1 or 2) corresponding to the value of the <i>sides</i> attribute of a document submitted to this printer.</p>

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	BSD option	Advanced Printing Software command, option, and value	Action
lpr	-P <i>destination</i>	pdpr -p <i>printer_name</i>	Submits the job to the specified logical printer.
lpr	# <i>number</i>	pdpr -n <i>copies</i>	Specifies the number of job copies you want printed.
lpr	-m	pdpr -N <i>email</i>	Delivers messages regarding the job by electronic mail.
lpr	-J <i>job</i>	pdpr -t <i>job_name</i>	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.
lpq	-P <i>destination</i>	pdq -p <i>printer_name</i>	Lists the status of print jobs for the specified printer.
lpq	-l	pdq -r all -s <i>line</i>	Lists information in long format, including the name of the host from which the print request originated.
lprm	<i>request_id</i>	pdrm <i>job_id</i>	Cancels the job specified by <i>job_id</i> .

Index

A

- abbreviating
 - attributes, 1-6
- attributes
 - abbreviating, 1-6
 - changing, 1-5
 - clearing, 1-5
 - default values, 1-5
 - defaults for printer, 2-4
 - defined, 1-2
 - document, 1-4, A-1
 - for logical printer, A-5
 - for printers, 2-3, 3-1, 3-12
 - for specific printer, 2-4
 - for text jobs, A-3
 - getting details, 4-2
 - job, 1-4, A-1
 - job scheduling, 3-5
 - overriding, 1-5
 - printer, 1-4, 3-7
 - storing in a file, 2-6, 3-9
 - text job, 1-4
 - values for, 1-5

B

- Benefits, 1-1
- BSD print system, 1-1, B-1

C

- canceling
 - print job, 4-6
- CDE, 1-2
- CLI, 1-1
- command equivalents
 - BSD system, B-1

- command line interface, 1-1
- commands
 - end user, 1-4
 - syntax for, 1-3

D

- default printer, 2-1
- discarding a job, 3-7
- document
 - attributes for, A-1
 - defined, 1-2
 - printing, 3-1
 - printing format, 3-3

E

- e-mail notification, 3-9
- end user
 - commands, 1-4
- environment variable
 - PDPRINTER, 2-1
- errors
 - print job, 4-2

F

- filter
 - modification, 3-10
 - translation, 3-10
- filters
 - for printing, 3-10
- finishing processes
 - specifying, 3-5

G

graphical user interface, 1-1, 1-2
GUI, 1-1, 1-2

H

help
for a command, 1-7
holding a job, 3-5

I

initial values
determining, 2-5
documents, 2-5
print jobs, 2-5
input trays
specifying, 3-4

J

job
attributes for, A-1
defined, 1-2
discarding, 3-7
errors, 4-2
retaining, 3-6
states, 4-1
submitting, 1-1
job identifier, 1-6
job scheduling
attributes for, 3-5

L

logical printer, 1-1, 2-1
attributes for, A-5
default, 2-2
defined, 1-2
specific, 2-2
lpd servers and printers, 1-1
lpr/lp commands, 1-1

M

man command, 1-7
media
specifying, 3-3
modifying
print job, 4-3
monitoring
print job, 1-1

O

object
defined, 1-2
operand, 1-4
option arguments, 1-3
options, 1-3
output bins
specifying, 3-4

P

pausing
print job, 4-4
pdl command, 1-4, 2-3, 2-4, A-6
pdmod command, 1-4, 1-5, 4-3
pdpause command, 1-4, 3-6, 4-4
pdpr command, 1-4, 2-2, 3-1, A-1
pdprint, 1-2
PDPRINTER
environment variable, 2-1
pdprintinfo, 1-2
pdq command, 1-4, 2-2, 4-1
pdresubmit command, 1-4, 4-5
pdresume command, 1-4, 3-6
pdrm command, 1-4, 4-6
pdset command, 1-5
physical device
defined, 1-2
physical printer, 1-1, 2-1
defined, 1-2
print job
canceling, 4-6
modifying a job, 4-3
monitoring, 1-1

- pausing, 4-4
- resubmitting, 4-5
- resuming, 4-4
- submitting, 1-1, 3-1

printer

- attributes, 3-12
- checking status, 4-1
- default, 2-1
- default settings, 3-7
- determining settings, 2-5
- getting specific attributes, 2-4
- initial values for documents, 2-5, 3-7, 3-8
- initial values for jobs, 2-5, 3-7, 3-8
- job errors, 4-2
- job states, 4-1
- logical, 1-1, 2-1
- physical, 1-1, 2-1
- properties, 2-3
- requesting attributes, 2-3
- setting default attributes, 2-4

printers

- on a spooler, 2-3

printing

- after a time, 3-6
- console message, 3-9
- discarding a job, 3-7
- document format for, 3-3
- e-mail notification, 3-9
- filters, 3-10, 3-11
- finishing processes, 3-5
- from standard input, 3-5
- holding a job, 3-5
- input trays, 3-4
- job status for, 3-9
- media for, 3-3
- number of copies, 3-2
- output bins, 3-4
- retaining a job, 3-6

- specifying attributes for, 3-1
- storing attributes for, 2-6, 3-9
- text to PostScript, 3-11
- to specific printer, 3-12
- two-sided, 3-2
- with pdpr command, 3-1

Q

- queue
 - defined, 1-2

R

- resubmitting
 - print job, 4-5
- resuming
 - print job, 4-4
- retaining a job, 3-6

S

- server
 - defined, 1-2
- spooler
 - printers for, 2-3
- status
 - checking printer, 4-1
- submitting
 - print job, 1-1, 3-1
- syntax
 - for commands, 1-3

T

- terminology, 1-1
- text job
 - attributes, A-3
- text to PostScript, 3-11