Xerox DocuPrint Network Printer Series Guide to Managing Print Jobs

THE DOCUMENT COMPANY
XEROX

Version 7.1
August 2000
721P87471

Xerox Corporation
701 S. Aviation Boulevard
El Segundo, CA 90245

Printed in the United States of America.

Publication number: 721P87471

Changes are periodically made to this document. Changes, technical inaccuracies, and typographic errors will be corrected in subsequent editions.

Laser safety

⚠ **Warning:** Adjustments, use of controls, or performance of procedures other than those specified herein may result in hazardous light exposure. ⚠

The Xerox DocuPrint printers are certified to comply with the performance standards of the U.S. Department of Health, Education, and Welfare for Class 1 laser products. Class 1 laser products do not emit hazardous radiation. The DocuPrint printers do not emit hazardous radiation because the laser beam is completely enclosed during all modes of customer operation.

The laser danger labels on the system are for Xerox service representatives and are on or near panels or shields that must be removed with a tool. **DO NOT REMOVE LABELED PANELS OR PANELS NEAR LABELS. ONLY XEROX SERVICE REPRESENTATIVES HAVE ACCESS TO THESE PANELS.**

Ozone information

This product produces ozone during normal operation. The amount of ozone produced depends on copy volume. Ozone is heavier than air. The environmental parameters specified in the Xerox installation instructions ensure that concentration levels are within safe limits. If you need additional information concerning ozone, call 1-800-828-6571 to request the Xerox publication 600P83222, *OZONE*.

Operation safety

Your Xerox equipment and supplies have been designed and tested to meet strict safety requirements. They have been approved by safety agencies, and they comply with environmental standards. Please observe the following precautions to ensure your continued safety.

- Always connect equipment to a properly grounded electrical outlet. If in doubt, have the outlet checked by a qualified electrician.

⚠ **Warning:** Improper connection of the equipment grounding conductor may result in risk of electrical shock.

- Never use a ground adapter plug to connect equipment to an electrical outlet that lacks a ground connection terminal.

- Always place equipment on a solid support surface with adequate strength for its weight.
- Always use materials and supplies specifically designed for your Xerox equipment. Use of unsuitable materials may result in poor performance and may create a hazardous situation.
- Never move either the printer or the Printer Controller without first contacting Xerox for approval.
- Never attempt any maintenance that is not specifically described in this documentation.
- Never remove any covers or guards that are fastened with screws. There are no operator-serviceable areas within these covers.
- Never override electrical or mechanical interlocks.
- Never use supplies or cleaning materials for other than their intended purposes. Keep all materials out of the reach of children.
- Never operate the equipment if you notice unusual noises or odors. Disconnect the power cord from the electrical outlet and call service to correct the problem.

If you need any additional safety information concerning the equipment or materials Xerox supplies, call Xerox Product Safety at the following toll-free number in the United States:

1-800-828-6571

For customers outside the United States, contact your local Xerox operating company.

Table of contents

Introduction

This *Xerox DocuPrint Network Printer Series Guide to Managing Print Jobs* contains information on print job specifications and print queue management.

About this guide

This guide is intended for system administrators or operators who are responsible for managing print jobs for the DocuPrint NPS. This guide is also intended for end users who are creating jobs with consideration to print management.

Users should have an understanding of Printer Controller operations and, for some tasks, be familiar with basic UNIX commands.

For more information on DocuPrint NPS tasks, refer to the complete *Xerox DocuPrint Network Printer Series* documentation.

Before using this guide, become familiar with its contents and conventions.

Contents

This section lists the contents of this guide:

- Chapter 1, “Document attributes,” provides information on how to change and view document attributes.
- Chapter 2, “Managing virtual printers,” contains information on how to set up and use virtual printers.
- Chapter 3, “Managing jobs,” describes the commands used to handle client print jobs.
- Chapter 4, “Specifying media,” provides information on how to set media specifications and how to use the setpagedevice media matching. This section also describes media limitations.
- Chapter 5, “Specifying multiset print jobs,” describes the commands that control multiset print jobs.
- Chapter 6, “Creating print jobs with referenced content,” provides information on how to manage referenced content files.
- Chapter 7, “Mixed plex applications,” describes DocuPrint NPS capability for setting up applications so that different plex values are used within a document.

Conventions

This guide uses the following conventions:

- All caps and angle brackets—Within procedures, the names of keys are shown in all caps within angle brackets (for example, press <RETURN>).
- Angle brackets—Variable information, or the position of a specified argument in the command syntax, appears in angle brackets (for example, List Fonts <Pattern>).
- **Bold**—Within procedures, text and numbers that you enter are shown in bold (for example, enter **privilege operator**).
- “Enter”—Within procedures, the two-step process of keying in data and pressing <RETURN> <ENTER> (for example, enter y).
- *Italics*—Document and library names are shown in italics (for example, the *Xerox DocuPrint Network Printer Series Guide to Managing Print Jobs*).
- Quotes—Keywords you can enter as arguments appear in quotes (for example, “USLetter”).
- Vertical bars—Alternatives to specified arguments are separated by vertical bars (for example, Set Time <Time | Remote Host Name | IP Address>).

Note: Notes are hints that help you perform a task or understand the text.

Caution: Cautions alert you to an action that could damage hardware or software. ⊘

Warning: Warnings alert you to conditions that may affect the safety of people. ⚠

Related publications

The Xerox DocuPrint Network Printer Series includes the following documents:

Decomposition Service and Tools Guide

Guide to Configuring and Managing the System

Guide to Managing Print Jobs

Guide to Performing Routine Maintenance

Guide to Submitting Jobs from the Client

Guide to Using Page Description Language

Installation Planning Guide

Messages Guide

System Overview Guide

Troubleshooting Guide

Glossary

Master Index

Customer Information Quick Reference Card

Printer Controller Commands Quick Reference Card

Submitting your Jobs from Macintosh Quick Reference Card

Submitting your Jobs from UNIX & DOS Quick Reference Card

Submitting your Jobs from Windows NT 4.0 (QuickPrint) Quick Reference Card

Submitting your Jobs Using Windows NT 4.0 Drivers Quick Reference Card

The documentation set also includes an electronic version, the *DocuPrint NPS Interactive Customer Documentation CD*.

1. Document attributes

The printing options requested by the user, or applied by a virtual printer, are examples of document attributes. In order for you to effectively manage jobs on your DocuPrint NPS, it is important for you to understand attributes, how they are set, and how they are used by the system.

This chapter describes how the system uses document attributes, lists values for attributes, and provides procedures to display and change attributes.

Understanding document attributes

DocuPrint NPS uses document attributes to track and process print requests. Before printing a job, DocuPrint NPS must have information about the desired printing characteristics for the job. Attributes are used as a means of providing that information, for example, media or highlight color to use, as well as other statistics such as sender name, job status, and so forth.

Attributes include the following characteristics of a document:

- Document name
- Sender and recipient of the document
- Printer to which the document is submitted and the print resolution
- Page ranges and the number of copies to print
- Order in which pages are printed
- Size, type, color, and weight of the media
- Number of sides of the paper to print on
- Finishing instructions, such as whether to staple and which output bin to use
- Priority of the job
- Information printed on the header page that prints with the job and messages that print at the end of a job
- Orientation and placement of the image on the page for ASCII documents
- Font name and size used for printing ASCII documents
- Information about how the job was completed

An attribute has a name, such as “CopyCount,” and a value, such as “2”, for printing two copies of the job. Refer to the section “Attributes and values” for a detailed list.

Setting attribute values

There are several ways attribute values can be assigned. Many attributes, especially those specifying print instructions, are set at the client workstation, either explicitly by the user or through the application being used. Attributes can be applied by the specified virtual printer. Some attributes have a default value assigned by the system if they are not explicitly set within the document itself. The defaults are defined with the commands explained in the Guide to Configuring and Managing the System.

You can display and change some document attributes after the document has been sent to the printer using the following commands:

- Change Document Attributes—Allows you to change many attributes of a document; however, it cannot be used to change the attributes of any job showing a “Printing” status in List Documents.
- Show Document Attributes—Lists all attributes of a document sent to the printer

These commands are described later in this chapter.

Methods for selecting DocuPrint NPS attributes

This section provides a brief description of methods used to set attributes. The following table indicates which attributes can be set by each method. Note that there are more attributes than those listed here. Refer to the section “Attributes and values” for detailed information on a specific attribute.

Note: The attribute name used by the system may have a somewhat different name on a client. For example, the attribute “CopyCount” is set by the Xerox job ticket “Copies” field. The “MapColor” attribute may be called “Extraction color” by some clients.

Xerox Job Ticket

The following are examples of job submission methods that support the Xerox Job Ticket (XJT):

- DocuPrint Submission Client for Windows platforms and Macintosh (XDS)
- Xerox Client “print” command for DOS and Unix
- DocuPrint NPS printer driver for Windows 95/NT.

The XJT is usually prepended to the file, and can be recognized by the characters %XRX at the beginning. Even though the XJT takes the form of PostScript comments, it is used for submission of PCL and ASCII files as well.

Virtual printer Refers to a logical printer that can have an associated set of document attributes that are applied to jobs submitted to it. If the job ticket or submission protocol already specifies the attribute, the virtual printer attribute is not used. Refer to the chapter “Managing virtual printers.”

lpr Refers to a job submission protocol. The protocol specifies the DocumentName and SenderName attributes. By using the -C option, the user can specify a wide range of attributes. Refer to the *Guide to Submitting Jobs from the Client*.

Netware PSERVER protocol Refers to a Novell NetWare protocol. The protocol specifies the DocumentName and SenderName attributes. Refer to the *Guide to Configuring and Managing the System* chapter “DocuPrint NPS on a Novell network.”

DSC Refers to Document Structuring Conventions that specify the format and comments of the PostScript program. These may be added by the application’s PostScript print driver, or they can be coded in the job by a PostScript programmer.

User-settable default Refers to a system default that can be defined by an NPS administrator. Refer to the *Guide to Configuring and Managing the System* chapter “Setting defaults.”

Always gets set Refers to attributes that are required, and are set either with values associated with the job, or by using system defaults.

Change Document Attributes Refers to attributes that can be modified by the operator using the Change Document Attributes command.

Table 1-1. **Methods of setting DocuPrint NPS attributes, examples**

Document Attribute	Xerox Job Ticket	Virtual Printer	lpr	Netware PServer Protocol	DSC	User-settable default	Always gets set	Change Document Attributes
AccountInfo	X	X	X					X
ASCIIFontName	X	X	X		N/A			X
ASCIIFontSize	X	X	X		N/A			X
AutoTabShift	X	X						X
BackgroundForm	X	X	X		X			X

[1] The DSC prescanner also picks up certain comments added via drivers using an Adobe PPD file.

[2] The user-selectable default, DefaultPSLanguageLevel only applies to PostScript jobs.

[3] The Plex attribute is not set when DefaultPlex is applied to a job.

[4] ForceBlackOnly will cause the HighlightColor to be set to Black.

[5] The PlexChangesDynamic attribute is not set when DefaultPlexChangesDynamic is applied to a job.

N/A- Not Applicable

Table 1-1. **Methods of setting DocuPrint NPS attributes, examples** (continued)

Document Attribute	Xerox Job Ticket	Virtual Printer	lpr	Netware PServer Protocol	DSC	User-settable default	Always gets set	Change Document Attributes
CopyCount	X	X	X		X		X	X
CycleForms		X	X					X
Disposition	X	X	X					X
DocumentName	X		X	X	X		X	
DocumentFormat	X	X	X		X	[2]	X	X
ElideHeader		X	X	X				X
ExceptionPages	X							X
Halftone	X	X	X					X
HighlightColor	X	X	X		[1]	[4]		X
HighlightColorRendering	X	X	X		[1]			X
HighlightColorMismatchAction	X	X	X					X
Hold		X	X					
HorizontalImageShift	X	X	X					X
MapColor	X	X	X		[1]			X
Media	X	X	X		X[1]	X	X	X
Orientation	X	X	X		N/A			X
OutputBin		X	X			X		X
PageOrder		X	X					X
PageRange	X	X						X
Plex	X	X	X		X[1]	[3]		X
PlexChangesDynamic	X	X	X			[5]		X
Priority		X					X	X
RecipientName	X	X						X
Resolution		X	X		[1]	X	X	X
SenderName	X		X	X	X[1]		X	

[1] The DSC prescanner also picks up certain comments added via drivers using an Adobe PPD file.

[2] The user-selectable default, DefaultPSLanguageLevel only applies to PostScript jobs.

[3] The Plex attribute is not set when DefaultPlex is applied to a job.

[4] ForceBlackOnly will cause the HighlightColor to be set to Black.

[5] The PlexChangesDynamic attribute is not set when DefaultPlexChangesDynamic is applied to a job.

N/A- Not Applicable

Table 1-1. **Methods of setting DocuPrint NPS attributes, examples** (continued)

Document Attribute	Xerox Job Ticket	Virtual Printer	lpr	Netware PServer Protocol	DSC	User-settable default	Always gets set	Change Document Attributes
SkipJTPrescan	N/A	X						
Staple	X	X	X		X[1]			X
Thicken		X	X					X
VerticalImageShift	X	X	X					X

[1] The DSC prescanner also picks up certain comments added via drivers using an Adobe PPD file.

[2] The user-selectable default, DefaultPSLanguageLevel only applies to PostScript jobs.

[3] The Plex attribute is not set when DefaultPlex is applied to a job.

[4] ForceBlackOnly will cause the HighlightColor to be set to Black.

[5] The PlexChangesDynamic attribute is not set when DefaultPlexChangesDynamic is applied to a job.

N/A- Not Applicable

Priority in setting attributes

Because there are a number of ways that document attributes are set during job submission (for example, the Xerox Job Ticket and a virtual printer), a priority system is defined. In the case of conflicting attribute definitions, the priority determines which value is assigned to the attribute in question. The priority is:

1. Xerox Job Ticket (unless SkipJTPrescan attribute is set)
2. Network submission protocol (for example, lpr or Novell Pserver)
3. Virtual printer
4. DSC comments
5. System defaults.

Example If a user's job ticket specifies duplex printing on white drilled paper, and the virtual printer attributes specify simplex printing on pink paper, the job will print duplex on white drilled paper.

Note: If you use a Xerox driver to create a PDL (such as PostScript) file and then submit the resulting PDL file via a network submission protocol such as lpr, the attributes from the PDL file job ticket will override any conflicting attributes set with lpr options. For example, if the PDL file job ticket requests simplex printing, and you request duplex with lpr, the job will print simplex. If you need to create PDL files for later submission by another means, you may wish to use a driver that does not create a Xerox job ticket.

Overriding attributes of a job ticket

Generally, the attributes set by the Xerox job ticket or submission protocol override attributes set by other methods (for example, the virtual printer).

When using a job submission method that creates a Xerox job ticket, the job attributes are not used when the virtual printer has the attribute SkipJTPrescan set to “True.” In this case, the job ticket attributes are ignored, and attribute settings for the virtual printer are used.

Overriding attributes of a virtual printer

You can override most virtual printer attributes using a submission client or through the job submission protocol (for example, lpr). This allows you to set many attributes on a per-job basis, regardless of the attributes set for the virtual printer.

Note: Xerox job ticket attributes will override the virtual printer attributes if:

Xerox Job Ticket processing was enabled at installation

and

The SkipJTPrescan attribute on the virtual printer is not set to TRUE.

The following are examples of overriding virtual printer attributes.

Media type attribute

When you submit a job, you can specify a plain prefinish paper stock or media type to override the Media virtual printer attribute. This example shows the command line to specify plain stock when printing the PostScript document, test.ps:

```
print -p <virtual printer name> -prefinish plain
test.ps
```

Staple attribute

You can override the Staple virtual printer attribute by specifying “none” for stitching. (This feature is not available for Models 96, 4635, or 180.) This example shows the command line to override the Staple attribute when printing the PostScript document, test.ps:

```
print -p <virtual printer name> stitching none
test.ps
```

Attributes and values

This section identifies the attributes that can be displayed or changed at the printer controller. It identifies the name of the attribute, accepted values, and whether it can be set by the Change Document Attributes (Change) or Change Virtual Printer (VP) commands, or whether it can be displayed only with Show Document Attributes (Show). Note that there are other attributes used internally by the system but which are not displayed or altered by using these commands.

Note: Ranges are specified for some attributes. If you enter a value outside the specified range, it may be accepted by the DocuPrint user interface, but the illegal value will be ignored during processing and printing.

Accountinfo <account information>

Any account identification information that appears on the header page that prints with the job.

Command: Show, Change, VP

ASCIIFontName <font>

The name of the Type 1 PostScript font used for printing ASCII documents.

Command: Show, Change, VP

ASCIIFontSize <number>

The size of the font (in points) used for printing ASCII documents.

Command: Show, Change, VP

AutoTabShift <“true” | “false”>

Set to “True” when a document requires a mix of normal and tab stock, and there is no support for printing on the tabs in the creation application.

When the AutoTabShift attribute is “True”, and a job has mixed media sizes, then for any selected medium that contains the text string “tab” in the type field of the media specification, the page image is shifted by the difference between the selected medium width and the default medium width for the job. (On the front side of the sheet the page contents are shifted to the right; on the back side of a duplex sheet, the shift is to the left.)

Command: Show, Change, VP

BackgroundForm <formname>

The name of the Background Form to be merged with the job. The form must have been saved using the ByteCode format. “formname” must match the name listed by the List Forms command. Refer to the Decomposition Service and Tools Guide for more details.

Command: Show, Change, VP

Category <text>

A user-defined category.

Command: Show, Change, VP

CompletionDate <date>

Date document was completed.

Command: Show

CopiesDone <number>

How many copies of the document have printed.

Command: Show

CopyCount <number of copies>

How many copies of the document were specified.

Command: Show, Change, VP

CPUMilliseconds <number>

Amount of time used by the controller’s microprocessor to process the job.

Command: Show

CycleExceptions <integer>

For jobs that are made up of many smaller sets of a fixed number of pages, this attribute allows page exceptions to be replicated throughout the document without needing the page exceptions to be explicit beyond the first set of pages.

Set the integer to the number of pages that constitute a set within the job. If the integer is less than 1, then the attribute has no effect.

Command: Show, Change, VP

CycleForms <n>

For a single multi-page Background Form associated with a document. Allows you to specify how to cycle through the pages when you run out of background images before the end of the job. Refer to the Decomposition Service and Tools Guide for more details.

Command: Show, Change, VP

Disposition <“SaveMaskG4” | “SaveFormG4” | “SaveMaskBC” | “SaveFormBC”>

Specifies the processing method used to create stored forms. The forms and masks can be saved electronically in BC or G4 formats:

– G4 - Saves each page of the job as separate 1-bit-per-pixel CCITT Group 4 compressed TIFF files.

– BC - Saves the jobs in the Xerox proprietary format ByteCode. It consumes a little more disk space; however, in some applications it is faster than SaveMaskG4. This format must be used to create forms that will by requested by the Background Forms attribute.

Mask - White pixels are transparent.

Form - White pixels are opaque.

Refer to the Decomposition Service and Tools Guide for more details.

Command: Show, Change, VP

DocumentEndMessage <text>

The message that appears on the Job Message sheet that prints at the end of a job.

Command: Show, VP

DocumentFormat <“PS”| “PS1” | “PS2” |“PCL” | “PCL5e” | “PCL5c” | “ASCII”>

Format of the document—PostScript with no level specified, PostScript level 1, PostScript level 2, PCL, PCL 5c, PCL 5e, or ASCII. “PS2” sends a file through a level 2 interpreter. Refer to the Set Option command for setting the default. PostScript language level. (The default at installation is 2).

Command: Show, Change, VP

DocumentID <number>

Identification number for the job; commonly referred to as “job ID”.

Command: Show

DocumentName <text>

The name of the document.

Command: Show

ElideHeader <“true” | “false”>

Allows suppression of header pages within a group of consecutive documents sent by the same SenderName attribute. If the printer cycles down between jobs, the header page will be printed regardless of this setting.

Command: Show, Change, VP

ExceptionPages <startpage, endpage, media | plex | imageShift | imageShiftForBack>

This attribute can be used to specify exception pages for media changes, plex changes, changes in vertical and horizontal image shift for the front pages, and changes in vertical and horizontal image shift for the back pages. The syntax for each type of exception is presented separately. Multiple ExceptionPages entries are acceptable when each entry is enclosed in square brackets.

Command: Show, Change

ExceptionPages <startpage, endpage, media>

Allows special media to be used for pages “startpage” through “endpage”. Media is specified as “size:type:color:weight.” Mixed paper sizes are supported.

When using Change Document Attributes to change the Exception Pages attribute, the Media attribute also needs to be changed if a new media is specified.

ExceptionPages <startpage, endpage,> plex, <plex value>

Allows plex change for pages “startpage” through “endpage”. Plex values can be: simplex, duplex, or tumbleDuplex.

ExceptionPages <startpage, endpage, shift type, shift value>

Allows change in image shift for pages “startpage” through “endpage”. The shift type can be HorizontalImageShift or VerticalImageShift for the front (or odd) pages, or HorizontalImageShiftForBack or VerticalImageShiftForBack for back (or even) pages of duplex jobs.

Shift value is expressed as an integer value in mm.

Note: The HorizontalImageShift and VerticalImageShift shift types apply to front pages only, regardless of whether the HorizontalImageShiftForBack or VerticalImageShiftForBack attributes are used.

Example The example shows using Change Document Attributes to change the ExceptionPages attribute for document 122. In the example there are multiple entries for changes to media, plex and shift type, each enclosed within square brackets. (For single entries, the square brackets are omitted.)

```
PS-Admin> Change Document Attribute 122 ExceptionPages
[1,2,UsLetter::blue:75],[3,4,Usletter::pink:75],
[5,23,UsLetter::white:75],
[1,2,plex,tumbleDuplex],[3,4,plex,simplex],
[5,23,plex,duplex],[1,2,horizontalimageshift,13],
[1,2,verticalimageshift,12]
```

FlipSide <“true” | “false”>

When true, sheets are turned over by the printer hardware before being sent to the stacker. The default is false.

Command: Show, Change, VP

Forwarded From <Hostname or IPAddress>

This is set when the “Forward Document” command was used on another DocuPrint controller to forward that document to the receiving DP controller.

Command: Show

Forwarded To <Host name or IP address>

This is set when the “Forward Document” command was used to forward that document from the forwarding DocuPrint controller.

Command: Show

Halftone <“defaultHalftone” | “coarse” | “medium” | “fine” | “extraFine”>

Use this command if you want to specify the way halftones will be printed in your PostScript or PCL job.

If you use the defaultHalftone value, the job’s halftone screen value will be used; it is equivalent to not specifying a value for the attribute. If you use one of the other values, the screen frequency listed below will be used. The default value for each system is indicated by an asterisk; the default is used where neither the job nor the halftone attribute specifies a screen value.

Table 1-2. **Halftone attribute values in lines per inch**

Printer	dpi	color	coarse	medium	fine	extraFine
4050/4090	300 dpi	b/w	42	53*	71	85
96/4635/180	300 dpi	b/w	42	53*	71	85
96/4635180	600 dpi	b/w	71	85	106*	141
4850/4890	300 dpi	b/w or h/c	42	53*	71	85
92C	300 dpi	b/w or h/c	42	53*	71	85
92C	600 dpi	h/c	53	71	85*	106
92C	600 dpi	b/w	71	85	106*	141

Command: Show, Change, VP

HighlightColor <“black” | “red” | “green” | “blue” | “cyan” | “magenta” | “cardinal” | “royalBlue” | “ruby” | “violet” | “brown” | “other”>

The highlight color used in the document (for highlight color models only).

Command: Show, Change, VP

HighlightColorMismatchAction <“abortPrint” | “useLoadedColor” | “requestOperatorIntervention”>

The action taken in the case of a toner color mismatch (for highlight color models only). A toner color mismatch occurs when the HighlightColor document attribute does not match the color loaded in the IOT.

Command: Show, Change, VP

HighlightColorRendering <“pictorial” | “presentation” | “colorToHighlight” | “colorTables” | “automatic” | “defaultRendering”>

The mapping of a three-dimensional color application (such as RGB) to the two-dimensional color application provided with highlight color printing (for highlight color models only). Because some information is lost in mapping, the color rendering attributes allow you to determine which qualities are the most important to preserve. The default setting is “automatic.”

pictorial

This attribute helps avoid color saturation problems with Decomposition Forms. Its name refers to maintaining the pictorial quality:

- Colors with a hue matching the map color are not changed.
- Colors with a hue close to the map color are rendered with color, but with a saturation level that decreases as the hue shifts away from the map color.
- Colors with a hue complimentary to the map color are produced in shades of gray.

presentation

This attribute distinguishes between different colors and is, therefore, a good choice for charts and diagrams as they typically use distinctly different colors. Colors that have a hue that matches the map color will not be changed. There will be discontinuities that may produce undesirable results in pictorial material.

colorToHighlight

This attribute discards the hue information and preserves the saturation and value of the color application. Fully-saturated input colors will turn into a solid highlight color despite the map color in effect. This is useful for documents that only use spot color for emphasis.

colorTables

This attribute uses preset color tables to perform the mapping and is provided for compatibility reasons. Color tables apply to red, green, and blue dry ink only.

automatic

This attribute chooses a mapping method on an object-by-object basis; for example, it applies pictorial mapping for pictures and presentation mapping for other objects (filled areas, strokes, and text). This is the default.

defaultRendering

This attribute uses the system default highlight color rendering attribute, set by using the Set Option command, to perform the mapping.

Command: Show, Change, VP

Hold

When true, this attribute will cause a job to be held.

Command: VP

Hold Reason

States the reason the job was held, if a reason was entered.

Command: Show

HorizontalImageShift <integer>

The placement of the image, in millimeters, from the edge of the page. This is used to reposition the printed image to allow extra space on the paper for binding. The shift is defined with respect to the orientation implied by the width and height of the media attribute, and does not depend on how the paper moves through the printer. For example, for USLetter or A4 paper with portrait orientation, on the front of the paper, a positive xshift will shift the image to the right. For duplex, the binding edge of the back side is on the left, so a positive xshift will move the image to the left.

Note: The value specified for this attribute will apply to both sides of a duplex page unless the HorizontalImageShiftForBack attribute is used for the back page.

Command: Show, Change, VP

HorizontalImageShiftForBack <integer>

The placement of the image, in millimeters, from the edge of the back (or even) page of a duplex job. The default used when a value for this attribute is not specified is 0 (zero) mm. This attribute can be overridden by the ExceptionPages shift attribute.

Note: If the HorizontalImageShiftForBack attribute is not set for a duplex job, the HorizontalImageShift value is used for both sides of the job.

Command: Show, Change, VP

HostName <name>

Name of the host from which the job was sent; used only when the job is submitting from lpr.

Command: Show

MapColor <"red" | "green" | "blue" | "cyan" | "magenta" | "yellow" | "cardinal" | "royal" | "ruby" | "violet" | "brown" | "black" | "other">

MapColor is the color that is replaced by the color specified in the HighlightColor document attribute (for highlight color models only), so that the highlight color is "mapped" onto the map color.

Note: Black can be used for the map color option where the entire document is in black, and you wish the content to be mapped into the highlight color. If you use this option for documents containing color, results are unpredictable and likely to be unacceptable.

Command: Show, Change, VP

Media <size:type:color:weight>

The media used for printing the document.

Acceptable size values for DocuPrint Models 4050, 4090, 4850, 4890, or 92C are "USLetter," "USLegal," "A4," and "nxn." Custom media sizes (nxn specified in millimeters) range from 8" x 10" (203 mm x 254 mm) to 8.5" x 14" (216 mm x 356 mm).

Acceptable size values for DocuPrint Models 96, 4635, and 180 are "USLetter," "USLegal," "USLedger," "Tabloid," "A3," "A4," and "nxn." Custom media sizes (nxn specified in millimeters) range from 7" x 10" (178 mm x 254 mm) to 17" x 14" (432 mm x 356 mm).

Use a text string to indicate the media type and color. When you specify a Media type from a client and you use the prefinish value, the word "plain" is equivalent to a blank media type.

You must specify the weight in grams per square meter (gsm).

The Media attribute can be assigned multiple values in a comma-separated list, for example:
USLetter::white,USLetter::blue,A4::white

Command: Show, Change, VP

MICRMode <“true” | “false”>

Applies to DocuPrint 96, 4635, and 180 MICR NPS systems only. When set to “true,” the printer runs in a MICR mode. This turns on the MICR mode monitor display icon, disables lead-edge-lightening, and resets the contrast control. The default is “false.”

ObjectSize

Size of the document in bytes. Does not include the size of any variable data or forms called externally from the document.

Command: Show

OffsetPerCopy <“true” | “false”>

Refers to whether the stacker offsets documents per copy when the OffsetPerCopy attribute is not specified. The default setting is “false.”

Command: Show, Change, VP

OffSetPerJob < “true” | “false”>

Refers to controlling job offsetting. Selecting “false” will disable job offsetting if the OffsetPerCopy option and attribute are also False and the OffsetPerCopy attribute is not set and the jobs do not request offsetting.

Command: Show, Change, VP

Orientation <“portrait” | “landscape” | “inversePortrait” | “inverseLandscape”>

The orientation of the ASCII document—portrait, landscape, inverse portrait, or inverse landscape.

Command: Show, Change, VP

OutputBin <“256” | “257” | “258” > or <“0” | ”1” | ”2”>

The output bin used for the document. Enter “0” or “256” to specify the sample tray (top tray), “1” or “257” to specify stacker group 1, and “2” or “258” to specify stacker group 2.

The stacker tray output bins are not set by default to be individually selectable. If you enter a stacker group as the output bin number, documents are sent to the stacker tray bin available for receiving output. If one bin fills to capacity, the system automatically outputs to the other bin.

The dual bin stacker tray output bins on Models 4050, 4090, 4850, 4890, or 92C can be reset by a service representative to be individually selectable. The service representative does this by changing the NVM settings on the printer. If the stacker trays are reset you could, for example, use the top output bin for one type of job and the bottom output bin for another. If you enter “257” as the output bin number, documents are only sent to the top bin. If you enter “258” as the output bin number, documents are only sent to the bottom bin. If one bin fills to capacity, the system does not automatically output to the other bin.

If you have the bypass transport option, which enables paper to pass through the printer output device to third-party finishing equipment, specify “257” as the output bin number for Models 4050, 4090, 4850, 4890, 92C.

The Model 96, 4635, and 180 output bins are:

256—top tray

257—high capacity stacker

258—bypass transport

Command: Show, Change, VP

PageOrder <“unspecified” | “ascend” | “descend” | “special”>

The order in which pages are printed—ascending, descending, or special.

Note: “Special” PageOrder tells the DocuPrint not to change the page order as it prints the job. If PageOrder is “Special”, and stapling is requested, the DocuPrint prints in the order presented, but face-up and stapled. This will appear to be reverse page order from what was presented.

Command: Show, Change, VP

PageRange <first page number-last page number>

The range of pages of the document that are printed.

Command: Show, Change, VP

PagesDone <number>

Number of pages completed. Note that this number does not directly correspond to the data kept by the meter. While a job is still printing, the number value for PagesDone will be higher than the actual number of pages currently delivered to the stacker.

Command: Show

PagesPerCopy <number>

Number of pages in each copy.

Command: Show

PagesPrinted <number>

Number of pages actually printed. Note that this number does not directly correspond to the data kept by the meter.

Command: Show

Plex <“simplex” | “duplex” | “tumbleduplex”>

Whether the document prints simplex, duplex, or tumble duplex.

Command: Show, Change, VP

PlexChangesDynamic <“true” | ”false”>

Set to “True” when you want to honor the plex changes specified in the job’s PostScript or PCL. Set to “False” when you want to ignore the plex changes specified in the job’s PostScript or PCL; the value used for the plex attribute will override any plex changes in the job. You can use the ExceptionPages attribute to make plex changes within the job.

Note: PCL front-side and back-side commands take effect for both the “True” and “False” values.

Command: Show, Change, VP

PrinterName <virtual printer name>

The name of the virtual printer to which the document was submitted. Virtual printer names must be unique within the network environment where they are running.

Use Change Document Attributes to change the PrinterName attribute from a stopped virtual printer to a started virtual printer.

Note: Changing the PrinterName attribute of a document does not affect other attributes assigned to the document; that is, the document will not inherit virtual printer attributes from another virtual printer.

Command: Show, Change

Priority <priority number>

The job priority. A low number indicates a low priority; a high number indicates a high priority. The range is 1 to 1000.

Command: Show, Change, VP

QuotaPages <number>

The maximum number of pages (not including header and trailer) of a document that can print before the document is cancelled by the system. This is an approximate number of pages.

Command: Show, Change, VP

QuotaSeconds <number>

The maximum number of seconds of CPU time that may be used in decomposing a document before the document is cancelled by the system. This is an approximate time.

Command: Show, Change, VP

ReadyToPrint <true | false >

Command: Show

RecipientName <job recipient name>

The name of the job recipient that appears on the header page that prints with the job.

Command: Show, Change, VP

ReprocessForCopies <“true” | “false”>

This attribute is supported only on the 96, 4635, and 180 NPS models with release 7.0 software.

Controls the use of reserve pages for multiple copies.

If set to “True,” then multiple copies are made by making multiple passes over the input. If set to “False” then multiple copies are made using the reserve pages as much as possible. When set to “False,” if a job runs out of reserve pages, the reserve pages are discarded and the job behaves as if set to “True.” The default is “False.”

For some multiple copy jobs, the system may achieve faster processing by not using reserve pages.

Command: Show, Change, VP

Resolution <“240” | “300” | “600”>

The printing resolution in dots per inch (dpi). 600 dpi is available for Models 92C, 96, 4635, and180 only. 240 dpi is available for the Model 180 only, and requires a special board.

On the 96, 4635, 180, and 92C NPS, output resolution is always 600 dpi regardless of the resolution of the input. For these models, if the specified value for the resolution attributes is 300 dpi, the data is decomposed by the controller at 300 dpi, and the print engine interpolates the data to 600 dpi; this provides the fastest processing. When the specified value for the resolution attribute is 600 dpi, the data is decomposed by the controller at 600 dpi; this provides the best quality image.

Command: Show, Change, VP

RetryCount <integer>

The number of attempts made to print a job. The RetryCount increases by one each time the sequencer is restarted while the job is printing. The “sequencer.attempts” parameter assigned at installation determines the maximum RetryCount before a job is held. Change Document Attributes can be used to raise or lower the number of retries before the job is held. If the “sequencer.attempts” parameter is set to 0, the RetryCount attribute has no effect.

Command: Show, Change

SenderName <sender name>

The name of the user sending the job.

Command: Show

SkipJTPrescan <“true” | “false”>

If true, the Xerox Job Ticket will not be used to set attributes on the job. If false, the Xerox Job Ticket will be used.

Command: Show, VP

Staple <“noStapling” | “singlePortraitStapling” | “otherstapling”>

Specifies whether the document is stapled after printing. This attribute is ignored on Models 96, 4635 or 180 without the stitcher output device.

Command: Show, Change, VP

Status <value>

The job’s status.

Command: Show

SubmissionDate <date>

Date the job was submitted.

Command: Show

SubmissionProtocol <protocol>

Network protocol used to submit the job.

Command: Show

Thicken <“0” | “1” | “10” | “11”>

The application of the thicken feature, which darkens each mark on the page. This attribute is valid only for the 4050 and 4090 models. The number refers to the imager parameter value as follows:

0 = do not darken

1 = darken all text and images

10 = darken bitmaps but not the text/graphics

11 = darken all text and images.

Command: Show, Change, VP

Uncollated <“true” | “false”>

If the value is set to “True,” then copies of jobs are not collated in the output. For example, if you print three copies of a three-page job, and Uncollated is set to True, the pages will print in the following sequence: 1,1,1,2,2,2,3,3,3. If the value is set to “False,” then copies of the jobs are collated in the output. In the example, when the value is False, the sequence for the pages will be: 1,2,3,1,2,3,1,2,3. The default is False, that is, copies will be collated.

Note: When using the Uncollated attribute keep in mind the following points:

- Copies of multiset sets and runs are collated or uncollated depending on the value of this attribute, but copies of the entire job are collated.
- Copies of individual pages are collated or uncollated when using the PCL copy command within the job, but copies of the entire job are collated.
- OffsetPerCopy does not apply when copies are uncollated.
- When stapling is requested, copies are not stapled when they are uncollated, though the job still prints face-up.

Command: Show, Change, VP

VerticalImageShift <integer>

The placement of the image, in millimeters, from the bottom of the page. This is used to reposition the printed image to allow extra space on the paper for binding. The shift is defined with respect to the orientation implied by the width and height of the media attribute, and does not depend on how the paper moves through the printer. For example, for USLetter or A4 paper with portrait orientation, on the front of the paper, a positive yshift will shift the image to the top. For tumble duplex, a positive yshift will move the back side image downwards.

Note: The value specified for this attribute will apply to both sides of a duplex page unless the VerticalImageShiftForBack attribute is used for the back page.

Command: Show, Change, VP

VerticalImageShiftForBack <integer>

The placement of the image, in millimeters, from the bottom of the back (or even) page of a duplex job. The default used when a value for this attribute is not specified is 0 (zero) mm. This attribute can be overridden by the ExceptionPages shift attribute.

Note: If the VerticalImageShiftForBack attribute is not set for a duplex job, the VerticalImageShift value is used for both sides of the job.

Command: Show, Change, VP

Change Document Attributes

Use the Change Document Attributes to change the attributes of a document that a client has sent to the printer.

This command cannot be used to change the attributes of any job with a status of “Printing.”

Access Level Administrator or Operator

Syntax Change Document Attributes <DocID(s)> <Attribute> <Value>

Arguments <DocID(s)>

A single document ID number or range of IDs or “all”. Use the List Documents command to determine the appropriate ID number.

<Attribute> <Value>

You can specify the following attributes and associated value. Refer to the section “Attributes and values” for detailed information on the values.

Note: For any argument that accepts text for the attribute value, for example, AccountInfo, only one word is accepted. If a multi-word phrase is entered, enclose it in double quotes.

AccountInfo <account information>

ASCIIFontName <font>

ASCIIFontSize <number>

AutoTabShift <“true” | “false”>

BackgroundForm <formname>

Category <text>

CopyCount <number of copies>

CycleExceptions <integer>

CycleForms <n>

Disposition <“SaveMaskG4” | “SaveFormG4” | “SaveMaskBC” | “SaveFormBC”>

DocumentFormat <“PS” | “PS1” | “PS2” | “PCL” | ”PCL5e” | “PCL5c” | “ASCII”>

ElideHeader <“true” | “false”>

ExceptionPages <startpage, endpage, media | plex | imageShift | imageShiftForBack>

FlipSide <“true” | “false”>

Halftone <“defaultHalftone” | “coarse” | “medium” | “fine” | “extraFine”>

HighlightColor <“black” | “red” | “green” | “blue” | “cyan” | “magenta” | “cardinal” | “royalBlue” | “ruby” | “violet” | “brown” | “other”>

HighlightColorMismatchAction <“abortPrint” | “useLoadedColor” | “requestOperatorIntervention”>

HighlightColorRendering <“pictorial” | “presentation” | “colorToHighlight” | “colorTables” | “automatic” | “defaultRendering”>

HorizontalImageShift <integer>

HorizontalImageShiftForBack <integer>

MapColor <“red” | “green” | “blue” | “cyan” | “magenta” | “yellow” | “cardinal” | “royalBlue” | “ruby” | “violet” | “brown” | “black” | “other” >

Media <size:type:color:weight>

OffsetPerCopy <“true” | “false”>

OffsetPerJob <“true” | “false”>

Orientation <“portrait” | “landscape” | “inversePortrait” | “inverseLandscape”>

OutputBin <“256” | “257” | “258” > or <“0” | ”1” | ”2”>

PageOrder <“unspecified” | “ascend” | “descend” | “special”>

PageRange <first page number—last page number>

Plex <“simplex” | “duplex” | “tumbleduplex”>

PlexChangesDynamic <“true” | “false”>

PrinterName <virtual printer name>

Priority <priority number>

QuotaPages <number>

QuotaSeconds <number>

RecipientName <job recipient name>

ReprocessForCopies <“true” | “false”>

Resolution <“240” | “300” | “600”>

RetryCount <integer>

Staple <“nostapling” | “singlePortraitstapling” | “otherstapling”>

Thicken <“0” | “1” | “10” | “11”>

Uncollated <“true” | “false”>

VerticalImageShift <integer>

VerticalImageShiftForBack <integer>

Example This example changes the copy count of document 60:

```
PS-op>change document attributes 60 copycount 10
Changing attribute copycount from 1 to 10.
```

Show Document Attributes

Use the Show Document Attributes command to show the attributes of a document sent to the printer from a client workstation.

Access Level Administrator, Operator, or User

Syntax Show Document Attributes <DocIDs or Custom Specification>

Arguments <DocID>

A single document ID number or range of ID numbers. Use the List Documents command to determine the appropriate ID number(s).

<Custom Specification>

You can use the following attributes to show specific documents.

Accountinfo <account information>

ASCIIFontName <font>

ASCIIFontSize <number>

AutoTabShift <“true” | “false”>

BackgroundForm <formname>

Category <text>

Completion Date <date>

CopyCount <number of copies>

CopiesDone <number>

CPUMilliseconds <number>

CycleExceptions <integer>

CycleForms <n>

Disposition <“SaveMaskG4” | “SaveFormG4” | “SaveMaskBC” | “SaveFormBC”>

DocumentEndMessage <text>

DocumentFormat <“PS” | “PS1” | “PS2” | “PCL” | ”PCL5e” | “PCL5c” | “ASCII”>

DocumentID <number>

DocumentName <text>

ElideHeader <“true” | “false”>

ExceptionPages <startpage, endpage, media | plex | imageShift | imageShiftForBack>

FlipSide <“true” | “false”>

Forwarded From <Hostname or IPAddress>

Forwarded To <Host name or IP address>

Halftone <“defaultHalftone” | “coarse” | “medium” | “fine” | “extraFine”>

HighlightColor <“black” | “red” | “green” | “blue” | “cyan” | “magenta” | “cardinal” | “royalBlue” | “ruby” | “violet” | “brown” | “other”>

HighlightColorMismatchAction <“abortPrint” | “useLoadedColor” | “requestOperatorIntervention”>

HighlightColorRendering <“pictorial” | “presentation” | “colorToHighlight” | “colorTables” | “automatic” | “defaultRendering”>

HoldReason <reason>

HorizontalImageShift <integer>

HorizontalImageShiftForBack <integer>

MapColor <“red” | “green” | “blue” | “cyan” | “magenta” | “yellow” | “cardinal” | “royal” | “ruby” | “violet” | “brown” | “black” | “other”>

Media <size:type:color:weight>

ObjectSize <size in bytes>

OffsetPerCopy <“true” | “false”>

OffsetPerJob <“true” | “false”>

Orientation <“portrait” | “landscape” | “inversePortrait” | “inverseLandscape”>

OutputBin <“256” | “257” | “258” > or <“0” | ”1” | ”2”>

PageOrder <“unspecified” | “ascend” | “descend” | “special”>

PageRange <first page number-last page number>

PagesDone <number>

PagesPerCopy <number>

PagesPrinted <number>

Plex <“simplex” | “duplex” | “tumbleduplex”>

PlexChangesDynamic <“true” | “false”>

PrinterName <virtual printer name>

Priority <priority number>

ProcessedByXIPP <“true” | “false”>

ProcessedByJTPrescan <“true” | “false”>

Promoted <“true” | “false”>

QuotaPages <number>

QuotaSeconds <number>

ReadyToPrint <“true” | “false”>

RecipientName <job recipient name>

ReprocessForCopies <“true” | “false”>

Resolution <“240”|“300” | “600”>

RetryCount <integer>

SenderName <sender name>

SkipJTPrescan <“true” | “false”>

Staple <“noStapling” | “singlePortraitStapling” | “otherstapling”>

Status <value>

SubmissionDate <date>

SubmissionProtocol <protocol>

Thicken <“0” | “1” | “10” | “11”>

Uncollated <“true” | “false”>

VerticalImageShift <integer>

VerticalImageShiftForBack <integer>

Example This example shows the attributes of document 22:

```
PS-op>show document attributes 22
Document ID: 22                   Sender: Michael Plass
Document Name:
/net/lane/byron/psprograms/RealWorld /SCOOP2.ps
Submitted: January 1, 1999 9:51 PST
Status: Complete, 1 Pages Printed
Media: USLetter, Duplex, Copies: 1
Priority: 500, Format: PS
Completion Date: January 1, 1998 9:52 PST
CPUMilliseconds: 5250,
Submission Protocol: lpd, Message: Sylvester
```

2. Managing virtual printers

Virtual printers are sets of document attributes that define the way in which a document will print. Operators can define virtual printers so that users do not have to explicitly set these attributes in each document they send to the printer. However, some of these attributes can be overridden by the attributes of a specific document; see "Overriding the attributes of a virtual printer" in the chapter "Document attributes."

Virtual printers that operators create are displayed by the DocuPrint NPS clients as follows:

- Xerox Windows Submission Client—as a queue name
- Macintosh client—in the Chooser window.

You manage virtual printers using the following commands:

- Change Virtual Printer—Assigns or changes attributes to a virtual printer.
- Create Virtual Printer—Creates a new virtual printer, but does not assign attributes. See Change Virtual Printer to assign attributes.
- Set Default Virtual Printer—Sets a default virtual printer so that a job that does not have a virtual printer specified at submission is associated with a virtual printer.
- Delete Virtual Printer—Deletes an existing virtual printer.
- List Virtual Printers—Lists all currently defined virtual printers.
- Start Virtual Printer—Starts an existing virtual printer to make jobs eligible for printing.
- Stop Virtual Printer—Stops an existing virtual printer to make newly submitted jobs ineligible for printing.
- Set Virtual Printer Flags—Exports or removes a virtual printer from AppleTalk.
- Copy Virtual Printer—Creates a virtual printer by copying the attributes of an existing virtual printer.

The following sections provide the syntax and parameters of these commands.

Change Virtual Printer

Use the Change Virtual Printer command to assign or change attributes to a virtual printer. Changing the attributes of a virtual printer does not affect the attributes of jobs that have already been submitted; it only affects the attributes of subsequent jobs.

Access Level Administrator or operator

Syntax Change Virtual Printer <Name> <Attribute> <Value>

Argument <Name>

The name of the virtual printer.

<Attribute> <Value>

You can specify the following attributes and associated values. Refer to the chapter “Document attributes” section “Attributes and values” for a detailed description of values.

Note: For any argument that accepts text for the attribute value, for example, AccountInfo, only one word is accepted. If a multi-word phrase is entered, enclose it in double quotes.

AccountInfo <account information>

ASCIIFontName <font>

ASCIIFontSize <number>

AutoTabShift <“true” | “false”>

BackgroundForm <formname>

Category <text>

CopyCount <number of copies>

CycleExceptions <integer>

CycleForms <n>

Disposition <“SaveMaskG4” | “SaveFormG4” | “SaveMaskBC” | “SaveFormBC”>

DocumentEndMessage <message>

DocumentFormat <“PS” | “PS1” | “PS2” | “PCL” | ”PCL5e” | “PCL5c” | “ASCII”>

ElideHeader <“true” | “false”>

Flipside <“true” | “false”>

Halftone <“defaultHalftone” | “coarse” | “medium” | “fine” | “extraFine”>

HighlightColor <“black” | “red” | “green” | “blue” | “cyan” | “magenta” | “cardinal” | “royalBlue” | “ruby” | “violet” | “brown” | “other”>

HighlightColorMismatchAction <“abortPrint” | “useLoadedColor” | “requestOperatorIntervention”>

HighlightColorRendering <“pictorial” | “presentation” | “colorToHighlight” | “colorTables” | “automatic” | “defaultRendering”>

Hold <“true” | “false”>

HorizontalImageShift <integer>

HorizontalImageShiftForBack <integer>

MapColor <“red” | “green” | “blue” | “cyan” | “magenta” | “yellow” | “cardinal” | “royalBlue” | “ruby” | “violet” | “brown” | “black” | “other”>

Media <size:type:color:weight>

MICRMode <“true” | “false”>

OffsetPerCopy <“true” | “false”>

OffsetPerJob <“true” | “false”>

Orientation <“portrait” | “landscape” | “inversePortrait” | “inverseLandscape”>

OutputBin <“256” | “257” | “258” > or <“0” | ”1” | ”2”>

PageOrder <“unspecified” | “ascend” | “descend” | “special”>

PageRange <first page number-last page number>

Plex <“simplex” | “duplex” | “tumbleduplex”>

PlexChangesDynamic <“true” | “false”>

Priority <priority number>

QuotaPages <number>

QuotaSeconds <number>

RecipientName <text>

ReprocessForCopies <“true” | “false”>

Resolution <“240” | “300” | “600”>

SkipJTPrescan <“true” | “false”>

Staple <“nostapling” | “singlePortraitstapling” | “otherstapling”>

Thicken = <“0” | “1” | “10” | “11”>

Uncollated <“true” | “false”>

VerticalImageShift <integer>

VerticalImageShiftForBack <integer>

Example

The following example assigns duplex printing to the virtual printer “psduplex”:

```
change virtual printer psduplex plex duplex
```

The following example shows how to specify the sample tray as the default output bin for a virtual printer called “ps1”:

```
change virtual printer ps1 outputbin 256
```

Create Virtual Printer

Use the Create Virtual Printer command to create a virtual printer.

Virtual printers that you create are displayed in the DocuPrint client printing protocols as follows:

- Xerox Windows Submission Client—as a queue name
- Macintosh client—in the Chooser window

Note: The Unix system administrator must manually set up the virtual printer name on the client. See the section on enabling lpr utilities in the *Xerox DocuPrint Network Printer Series Guide to Submitting Jobs from the Client*, “Setting up workstations.”

Note: Virtual printer names must be unique within the network environment where they are running.

Note: With AppleTalk you can only view 64 virtual printers per Printer Controller. You can create more than 64 virtual printers, but only 64 display in the Chooser menu.

If you create 65 virtual printers, the following message displays in the log, /var/log/DocuPrint:

> WARNING AppleTalk: Too many virtual printers - can’t add <printername>

The message does not display for any additional virtual printers. You can use the command Set Virtual Printer Flags to specify which virtual printers will be viewed in the Chooser.

Access Level Administrator or operator

Syntax Create Virtual Printer <Name>

Argument <Name>

Specifies the name of the new virtual printer.

Example This example shows how to create a virtual printer named ‘psduplex’:

```
PS-op>create virtual printer psduplex
Virtual printer psduplex created.
```

After you create the virtual printer, use the Change Virtual Printer command to define its attributes.

Copy Virtual Printer

Use the Copy Virtual Printer command to create a new virtual printer by copying a specific virtual printer and its attributes.

Access Level Administrator or operator

Syntax Copy Virtual Printer <existing> <new name>

Argument <existing>

Specifies the name of the existing virtual printer you want to copy.

<New virtual printer name>

Specifies the name of the new virtual printer that will be created using the attributes of the existing virtual printer.

Delete Virtual Printer

Use the Delete Virtual Printer command to delete a specific virtual printer. The system will not allow you to delete the default virtual printer.

Access Level Administrator or operator

Syntax Delete Virtual Printer <Name>

Argument <Name>

Specifies the name of the virtual printer you want to delete.

Example The following example deletes the virtual printer named “psduplex”:

```
PS-op>delete virtual printer psduplex
Virtual printer psduplex deleted.
```

List Virtual Printers

Use the List Virtual Printers command to display available virtual printers and their attributes.

Access Level Administrator, operator, or user

Syntax List Virtual Printers <Pattern>

Argument <Pattern>

Specifies the pattern of the virtual printer name you want listed. A specific name for one printer or a wildcard (*) for multiple printers may be used. If no pattern is specified, all virtual printers are listed.

Example The following example lists all virtual printers with “B” as the first letter in their name:

```
PS-op>list virtual printers b*
BACH

   Status:Started
   Attribute:PrinterName, BACH
   Attribute:Plex, duplex

BEETHOVEN

   Status:Started
   Attribute:PrinterName, BEETHOVEN
   Attribute:Media, USLegal::white
```

Print Virtual Printers List

Use the Print Virtual Printers List command to print a list of all or selected virtual printers and their associated attributes.

This command will also identify the default virtual printer (if * is the pattern or any pattern specified matches the default virtual printer name).

Access Level Administrator or operator

Syntax Print Virtual Printers List <Pattern>

Argument <Pattern>

Specifies the pattern of selected virtual printer names for a printed list. A printer may be selected by a combination of characters and the (*) wildcard. The (*) wildcard alone selects all virtual printers.

Example

The following example prints a list of all virtual printers with “M” as the first letter in their name:

```
PS-op>print virtual printers m*

Print Virtual Printers List at July 28, 1998
9:36:08 am PDT Submitted as document number 5
```

Set Default Virtual Printer

Use the Set Default Virtual Printer command to set the system default virtual printer.

The job pool manager allows the DocuPrint controller to assign attributes from a default virtual printer when no virtual printer was specified at submission.

Access Level

Administrator or operator

Syntax

Set Default Virtual Printer <Name>

Argument

<Name>

Specifies the name of an existing virtual printer to use as the system default virtual printer.

Example

This example shows how to set a default virtual printer named ‘psduplex’.

```
PS-op>set default virtual printer psduplex

 Default Virtual printer psduplex set.
```

Start Virtual Printer

Use the Start Virtual Printer command to start a virtual printer that has been stopped, making jobs submitted to the specified virtual printer eligible for printing. Virtual printers are started by default when the virtual printer is created by the Create Virtual Printer command.

Access Level

Administrator or operator

Syntax

Start Virtual Printer <Name>

Argument

<Name>

Specifies the name of the virtual printer you want to start.

Example

The following example starts the virtual printer called “psduplex”:

```
PS-op>start virtual printer psduplex
Virtual printer psduplex started.
```

Stop Virtual Printer

Use the Stop Virtual Printer command to stop a virtual printer, making jobs submitted to the specified virtual printer ineligible for printing.

Access Level

Administrator or operator

Syntax

Stop Virtual Printer <Name>

Argument

<Name>

Specifies the name of the virtual printer you want to stop.

Example

The following example stops the virtual printer called “psduplex”:

```
PS-op>stop virtual printer psduplex
Virtual printer psduplex stopped.
```

Set Virtual Printer Flags

Use the Set Virtual Printer Flags command to specify which virtual printers will be selected for viewing in the users' Macintosh Chooser.

DocuPrint NPS software limits the number of virtual printers that can be viewed in the Macintosh Chooser to 64 virtual printers from each NPS controller. This command is only needed if there are more than 64 virtual printers on a system, or if the system option ExportVPsToAppleTalk has been set to false so that newly created virtual printers are not automatically made available to AppleTalk. Refer to the *Guide to Configuring and Managing the System* chapter on "Setting System Defaults" for more information on the option.

Access Level

Administrator or operator

Syntax

Set Virtual Printer Flags <Name> <ExportToAppleTalk Flag>

Argument

<Name>

Name of the virtual printer that you want to make available for Macintosh users.

<ExportToAppleTalk Flag>

Enter True if you want the virtual printer to be made available in the Macintosh users' Chooser. Enter False if you wish to end the named virtual printer's availability to AppleTalk users.

```
PS-op>set virtual printer flags report_vp  true
```

Using Novell forms to specify virtual printers

If a Novell user wants to access a DocuPrint virtual printer for a PSERVER configuration, a form with the same name as the virtual printer must be defined by the NetWare administrator. Forms are defined on the Novell server using the Novell PRINTDEF program. The form name can have a maximum of twelve characters and the first character must be alphabetic. Use a prefix such as "DP" to distinguish these forms from other form names.

By default, the PServer configuration will use a virtual printer that has the same name as the Novell queue name. Other virtual printers can be specified using the form names.

Once the form is defined, the user must specify that form for the job.If a form name is not used, a virtual printer with the same name as the queue is used.

Note: DocuPrint NPS is not using a form in the Novell sense. It reads the name field of the form specified and interprets it as a DocuPrint virtual printer. None of the form attributes other than *name* is read.

3. Managing jobs

Print jobs are stored on the Printer Controller disk until a printer is ready. Multiple users can send jobs to the same printer at the same time. The software accepts and tracks jobs to enable users to view, remove, and change values of jobs before they are printed.

Each job has an assigned priority that DocuPrint uses, among other factors, to determine the order in which jobs print. DocuPrint provides certain guarantees about print job order. The following are print job conditions that relate to job order:

- Active and Held lists—Job order is shown by the output of the List Documents command (Active List). The job shown at the top of the Active List will be the first job printed (subject to Scheduling Policy).
- Insert order—A job is inserted into the Active List or the Held List by priority and then by DocID, but after all jobs of higher priority, jobs that were promoted, and jobs that were submitted by the service representative. Jobs that were promoted are marked by having the Promoted attribute set to TRUE. (Jobs that were removed from the Active List and placed in the Held List have the Promoted attribute set to FALSE.)
- Arriving jobs—Jobs appear in the Active List when they are being received. At the start of job arrival, the Job Pool Manager (JPM) sets the priority of the new job to the default priority.
- Priority—A maximum priority is enforced. A priority is an integer, with higher numbers corresponding to higher priorities. Highest priority is 1000 and lowest is 1. The default is 500.
- Reordering—Job position in the Active List changes when the priority of the job changes as a result of certain operator actions. Once the job is in the job pool, an operator may change the job priority. Jobs can also be promoted.
- Promote operation—Removes a job from its current position on either the Active List or Held List and inserts the job in the Active List, ahead of other pending jobs.
- Scheduling policies—You can choose between two policies (FirstComeFirstServed or ResourceMatch) for selection of the next print job. The Set Scheduling Policy command changes the policy. ResourceMatch is the system default, and allows the next job that is eligible (for example, requested media or toner color are loaded) to print. This may not be the next job in the job order. If FirstComeFirstServed is specified, then the system will print the jobs in order, even if jobs are ineligible and require operator intervention to load the requested resource.
- Hold operation—Removes a job from the Active List and inserts the job in the Held List according to insertion order. The job can later be released.

Note: Some operations allow you to specify a group of documents. You can specify the group by listing individual DocIDs, specifying DocID ranges, or both. In all cases, specifying a group of documents is treated as specifying an unordered set of jobs.

With these conditions in mind, you can manage the jobs using the following commands:

- Cancel Documents—Cancels a print job with a “held,” “pending,” or “ineligible” state. Attempts to cancel jobs that are currently printing.
- Forward Document—Transfers jobs from one DocuPrint to another DocuPrint.
- Hold Documents—Holds a specified document from being printed.
- List Documents—Lists documents that have a specified state or attribute.
- Promote Document—Moves a pending job ahead of other jobs that are pending.
- Release Documents—Releases a document in a hold status.
- Set Scheduling Policy—Sets the policy by which the system determines the next job to print.
- Show Scheduling Policy—Displays the current site scheduling policy.
- Start—Starts printing and queueing of jobs.
- Stop—Stops printing and queueing of jobs.

The following sections show the syntax and parameters of these commands. Refer to the *Troubleshooting Guide* for the List Sample Document and Print Sample Document commands.

The Change Document Attributes command can also be used to manage a job. Refer to the chapter “Document Attributes.”

All job management tasks are performed at the Printer Controller or issued through a remote telnet session.

Cancel Documents

Use the Cancel Documents command to cancel a print job while it is in the “held,” “pending,” or “ineligible” state. You can specify Cancel Documents for a job in the “printing” state, and the system will attempt to cancel it. You can cancel a single job or a range of documents.

Note: If the last page of the document has been decomposed, as many as 160 pages may be delivered to the output tray before a trailer sheet with the message “Error: job cancelled by user” is delivered. If a job has completed decomposing, use the Restart Sequencer command to change an active job status to “held” and then cancel the document.

Access Level Administrator or operator

Syntax Cancel Documents <DocID(s) | Custom Specification>

Arguments <DocID(s)>

A single number, a list of numbers or ranges separated by commas (for example, 2-5,7,9-15), or “All.”

Use the List Documents command to determine a document ID number.

<Custom Specification>

See “List Documents” for custom specification details.

Example The following example cancels documents 5 through 17:

```
PS-op>cancel documents 5-17

Do you really want to cancel documents 5-17? y
Documents 5-17 cancelled
```

Forward Document

Use the Forward Document command, available in Privilege Operator or Administrator mode, to transfer jobs from one DocuPrint to another that is running the same version of the software. Only documents with a Pending or Ineligible status can be forwarded. If you try to forward a document with another status, you receive the following message:

```
There are no documents to be forwarded.
```

After the document is forwarded to the new DocuPrint, you receive a message asking if the document should be cancelled on the forwarding DocuPrint.

If a virtual printer on the system to which the document is forwarded matches the PrinterName attribute for the document, the document will inherit attributes from the new virtual printer. If the new virtual printer is stopped, the forwarded document will have an Ineligible status.

Note: The Forward Document command will not automatically forward variable data or images external to the submitted PostScript file. You must ensure that these files are stored on the DocuPrint system that is receiving the forwarded document.

Note: The Forward Document command uses the TCP/IP protocol to transfer data. If you are not using TCP/IP on the local network, documents can still be forwarded to other systems on the same subnet as long as each system has been given a unique IP address. On such a network, documents can be forwarded to another DocuPrint on other subnets only if your network bridges and routers have been configured to pass IP packets.

The Forward Document command forwards jobs only to other DocuPrint NPS systems with the same version of NPS software installed.

Access Level Operator or administrator

Syntax Forward Document <DocID(s) | Custom Specification> <Host name | IP address>

Arguments <DocID(s)>

A single number, a list of numbers, or ranges separated by commas (for example, 2-5,7,9-15), or “All.”

Use the List Documents command to determine a document ID number.

<Custom Specification>

See “List Documents” for custom specification details.

<Host name>

The hostname of the receiving DocuPrint. If hostname is used in the command, the forwarding DocuPrint must be configured to recognize that hostname by having an appropriate entry in the /etc/hosts file. If DNS is configured, you can use the full domain path.

<IP address>

The IP address of the DocuPrint to receive the document.

Example The following example forwards Document 12146 to the host named atlas:

```
PS-Admin>forward document 12146 atlas

Do you really want to forward document 12146? y
Document 12146 forwarded to atlas as docID 4391.

Do you want to Cancel the document just forwarded
(12146)? y
```

Hold Documents

Use the Hold Documents command to hold specified documents. You can hold only those documents that are in the pending or ineligible states.

Access Level Administrator or operator

Syntax Hold Documents <DocID(s) <Reason>>

Arguments <DocID(s)>

A single number, a list of numbers, or ranges separated by commas (for example, 2-5,7,9-15), or “All.”

Use the List Documents command to determine a document ID number.

<Reason>

The reason the specified documents are being held (up to 600 characters).

Example This example holds job 5:

```
PS-op>hold documents 5
Document held: 5
```

List Documents

Use the List Documents command to list documents that have a specified status or attribute. If you do not specify any arguments, this command lists all the documents that are not complete. It will also indicate the number of documents that are in the “hold” state.

This command limits the total output to 100 characters on the DocuPrint Administration window, and 80 characters on Remote Administrations connections. Sender names and document file names may be truncated.

Note: When your document does not display with the List Documents command and it appears to be missing because it did not print, use the List Documents Held command. Refer to the <List Option> arguments shown below.

Access Level All

Syntax List Documents <List Option> <DocID(s) | Custom Specification>

Arguments <List Option>

The status of the documents you want to list.

- "All"
- "Complete"
- "Held"
- "Ineligible"—Lists documents that cannot be processed because they are in a "media wait" state, a "color housing wait" state, or were submitted to a virtual printer that has been stopped.

Documents in the "media wait" state are jobs that require a certain medium that is not present in the trays. Use the List Documents Ineligible command to determine the job requirement. Load the correct medium into the tray. Use the Set Tray command to notify the system which medium is in the tray.

Documents in the "color housing wait" state are waiting for a CCU change. Use the List Documents Ineligible command to determine the correct color requirement for the job. Change the color housing.

To print documents that were submitted to a stopped virtual printer, use the Start Virtual Printer command to restart the virtual printer.

The "All" or "Complete" options will display the number of pages printed. Cancelled documents will not list pages printed even if some pages did print before the job was cancelled.

<DocID(s)>

A single number, a list of numbers, or ranges separated by commas (for example, 2-5,7,9-15), or "All."

<Custom Specification>

To enter a custom specification use the format:

<attributeName> <"=" | "==" | "!="> <attributeValue>

The single or double equal sign ("=" or "==") can be used to signify "equal to" and the exclamation point followed by an equal sign ("!=")is used to signify "not equal to."

You can use the following attributes to list specific documents. See the chapter "Document attributes" section "Attributes and values" for detailed information on the values.

Accountinfo <account information>

ASCIIFontName <font>

ASCIIFontSize <number>

AutoTabShift <"true" | "false">

BackgroundForm <formname>

Category <text>

CompletionStatus <"complete" | "cancelledByOperator" | "cancelledByUser" | "completeWithWarnings" | "completeWithErrors" | "rejected" | "aborted">

CopyCount <number of copies>

CycleForms   <n>

Disposition <“SaveMaskG4” | “SaveFormG4” | “SaveMaskBC” | “SaveFormBC”>

DocumentEndMessage <text>

DocumentName <text>

DocumentFormat <“PS”| “PS1” | “PS2” |“PCL” | “ASCII” |“PCL5e” | “PCL5c”>

ElideHeader <“true” | “false”>

ExceptionPages <startpage, endpage, media>

Forwarded From <Hostname or IPAddress>

Forwarded To <Host name or IP address>

HighlightColor <“none” | “black” | “red” | “green” | “blue” | “cyan” | “magenta” | “cardinal” | “royal” | “ruby” | “violet” | “brown” | “other”>

HighlightColorMismatchAction <“abortPrint” | “useLoadedColor” | “requestOperatorIntervention”>

HighlightColorRendering <“pictorial” | “presentation” | “colorToHighlight” | “colorTables” | “automatic”>

HorizontalImageShift <integer>

MapColor <“none” | “red” | “green” | “blue” | “cyan” | “magenta” | “yellow” | “cardinal” | “royal” | “ruby” | “violet” | “brown” | “black” | “other”>

Media <size:type:color:weight>

Orientation <“portrait” | “landscape” | “inversePortrait” | “inverseLandscape”>

OutputBin <“256” | “257” | “258” > or <“0” | ”1” | ”2”>

PageOrder <“unspecified” | “ascend” | “descend” | “special”>

PageRange <first page number-last page number>

Plex <“simplex” | “duplex” | “tumbleduplex”>

PrinterName <virtual printer name>

Priority <priority number>

QuotaPages <number>

QuotaSeconds <number>

RecipientName <job recipient name>

Resolution <“240” | “300” | “600”>

RetryCount <integer>

SenderName <sender name>

SkipJTPrescan <“true” | “false”>

Staple <“noStapling” | “singlePortraitStapling” | ”otherstapling”>

Thicken <“0” | “1” | “10” | “11”>

VerticalImageShift = <integer>

Example This example lists all uncompleted documents:

```
PS-op>list documents
Doc #   Sender Name   Document Name    Status
    6   ISTL          Test Doc 2       Ineligible
    7   ISTL          Test Doc 3       Ineligible
    8   ISTL          Test Doc 4       Pending
    9   ISTL          Test Doc 5       Processing
4 Documents Listed
```

The following example lists all ineligible documents and the resources they require for “media wait,” “color housing wait,” and “stopped virtual printer”:

```
PS-op>list documents ineligible
Doc #    Sender Name    Document Name    Required Resource
 265     Administrator HC               Color=Blue
 266     Administrator 4pages           Media=UsLegal::white
 267     prtcon         xret11.ps        VP=roverq
3 Documents Listed
PS-admin>
```

Promote Document

Use the Promote Document command to take a job in the “pending” state and advance that job ahead of other jobs in the “pending” state. Promoting a document in the “ineligible” state does not force the document to print.

Promote Document does not change the priority attribute. Promoted jobs will print ahead of jobs with a priority of 1000 (the highest priority) regardless of the priority of the promoted job.

Access Level Administrator or operator

Syntax Promote Document <DocID>

Arguments <DocID>

The document ID number. You can use the List Documents command to determine a document ID number.

Example This example promotes job 8:

```
PS-op>promote document 8
Document 8 promoted
```

Release Documents

Use the Release Documents command to release for processing a document you have previously placed in a “held” state.

Access Level Administrator or operator

Syntax Release Documents <DocID(s) | Custom Specification>

Arguments <DocID(s)>

A single number, a list of numbers, or ranges separated by commas (for example, 2-5,7,9-15), or “All.”

You can use the List Documents command to determine a document ID number.

<Custom Specification>

See “List Documents” for custom specification details.

Example This example releases jobs 5, 8, and 12:

```
PS-op>release documents 5, 8, 12
Documents released: 5,8,12
```

Set Scheduling Policy

Use the Set Scheduling Policy command to set the policy by which the system determines the next job to print. A change to the policy takes place immediately.

You can choose between two policies (FirstComeFirstServed or ResourceMatch) for selection of the next print job. ResourceMatch is the system default, and allows the next job that is eligible (for example, requested media or toner color are loaded) to print. This may not be the next job in the job order, if the next job is ineligible. If FirstComeFirstServed is specified as the policy, then the system will print the jobs in absolute order, even if a job is ineligible and requires operator intervention to load the requested resource before printing continues.

Note: Using this command may change the order in which jobs print.

Access level Administrator or operator

Syntax Set Scheduling Policy <“resourceMatch” | “FirstComeFirstServed”>

Arguments

<resourceMatch>

Indicates that DocuPrint should print the first job in the Active list that is not already printing and for which all resources are available. Jobs that do not have necessary resources loaded become ineligible.

<FirstComeFirstServed>

Indicates that DocuPrint should print the next job in order. If the required resources are not available for this job, printing stops until the resources are available or the operator removes the job, or moves the job to the Held list.

Show Scheduling Policy

Use the Show Scheduling Policy command to display the site scheduling policy currently specified.

Access level

Administrator or operator

Syntax

Show Scheduling Policy

Parameters

None

Start

Use the Start command to start printing, queueing, or both.

Access Level

Administrator or operator

Syntax

Start <Printing | Queueing | All>

Arguments

<Printing>

Starts the printing of jobs in the queue.

<Queueing>

Allows the Printer Controller to receive new jobs from clients

<All>

Starts the printing of jobs in the queue and allows the Printer Controller to receive new jobs from clients.

Example

The following example starts both printing and queueing on the Printer Controller:

```
PS-op>start all

Normal Printing Started.
Queueing Started.
```

Stop

Use the Stop command to stop printing, queueing, or both.

Access Level

Administrator or operator

Syntax

Stop <Printing | Queueing | All>

Arguments

<Printing>

Stops pending jobs from printing. If the document has begun decomposing (status is “printing”), it cannot be stopped. If a job has begun decomposing, refer to the Restart Sequencer command to change an active job status to “held.”

For “held” or “received” jobs, refer to the Set Scheduling Policy command.

<Queueing>

Stops the Printer Controller from receiving new jobs from clients.

<All>

Stops the printing of pending jobs and prevents the Printer Controller from receiving new jobs from clients.

Example

The following example stops the Printer Controller from receiving jobs:

```
PS-op>stop queueing
Queueing Stopped.
```

Managing print jobs using Novell Netware

Novell NetWare software receives and queues print jobs. It assigns priorities to incoming jobs and maintains a temporary record of information for every print job. The information includes such items as job number, job position in the queue, time the job entered the queue, print server designated to print the job, userid of the person who submitted the job, and so on. The software also sends print jobs to the printer, and contains a series of utilities and menus to set up, manage, and manipulate print queues:

- PCONSOLE
- PSERVER
- RPRINTER

The following sections provide details about these utilities. Refer to the *Guide to Configuring and Managing the System* for more information.

PCONSOLE

PCONSOLE is the main Novell utility used for creating, viewing and managing print queues. When you create a queue with PCONSOLE, the Novell software sets up a queue directory and a bindery entry. Every queue has a directory on the hard disk. A print job sent to a queue is stored as a file within that directory. Once the job is printed, the file is removed from the queue directory. The bindery is the database of all users, groups, and other defined entities on the NetWare file server.

Note: There is no PCONSOLE access at the DocuPrint NPS Printer Controller.

You can perform the following tasks using PCONSOLE:

- Add or remove the following:
  - definitions of file servers serviced by a print server
  - definitions of print queues serviced by a print server
  - definitions of printers for a print server
  - print servers
  - jobs from a queue
  - queues
- View or change job information
- Check or change queue status.

A user operator can change job values or delete a job while it is still in the Novell queue and before DocuPrint NPS starts to receive it. If the user has deleted the job after DocuPrint NPS begins to receive it, DocuPrint receives the delete action and attempts to cancel the job. Cancellation may fail and the job will partially or completely print.

PSERVER

A Novell print server (PSERVER) is a combination of hardware and software that takes ready-to-print jobs from the print queue and sends them to the printer. The print server software is the link between the print queue and the printer (either local or remote). Only one print server can be defined on a file server. Each print server can service up to 16 printers. A print server can also run on a dedicated machine that is not running as a file server.

Note: Although Novell PSERVER can support multiple queues on multiple file servers, DocuPrint NPS PSERVER does not. It supports one queue on one file server only. A PSERVER must be created for each DocuPrint on the Novell file server. The PSERVER process is actually run on the DocuPrint Printer Controller. Do not load PSERVER processes on the Novell Server that connects to the DocuPrint.

The Novell PSERVER software scans the print queues to which it is assigned looking for jobs that are ready to print. When it obtains a job from a queue, the print server sends the data to the DocuPrint NPS through an NCP/IPX connection. Once the job is printed, it is deleted from the Novell print queue and the file is removed from the queue directory.

The DocuPrint PSERVER leaves the job in the queue and updates the queue entry as processing proceeds. The job is removed from the queue when it is done printing or is cancelled by an operator.

The following job parameters are affected when the DocuPrint NPS is configured as a PSERVER:

- Job description is used as the document name.
- Copies is used to set the CopyCount attribute.
- Sender name is set from the user name.
- A message is sent to the user when the job finishes printing or is canceled, if the Notify flag is set on the job and the user is logged into the file server.
- ElideHeaders is set to TRUE, if you enable the banner.
- Form name is used to specify the virtual printer, other than the default printer. See “Managing Virtual Printers” for more information. (For PSERVER, queue name is the default virtual printer name, if the Novell queue name is the same as one of the corresponding DocuPrint virtual printer names.)

RPRINTER

A Novell remote printer workstation is a vehicle for the print server to move print data to a printer that is not directly connected to the print server. The remote printer software normally runs on a workstation that connects to the printer through a parallel or serial port. DocuPrint NPS can function as an RPRINTER.

In this configuration, DocuPrint NPS communicates with a print server running print server software. The print server services jobs from the print queue and sends the data over the network through an SPX connection to the DocuPrint RPRINTER. There can be up to sixteen RPRINTERS serviced by the Novell Print Server (PSERVER). The DocuPrint links each of these remote printers to the DocuPrint virtual printer with the same name (if there is one). Because each of these remote printers can be defined to service a different queue (possibly on a different file server), a queue can be established for up to sixteen virtual printers.

Note: When DocuPrint NPS is connected as an RPRINTER, Novell banner pages must be disabled on all jobs.

The RPRINTER does not have access to the queue or the data for a job in the queue.

4. Specifying media

A media specification is a statement of the media required to print a job. It may consist of one or more of the following criteria: size, weight, color, and type.

You specify media through static and dynamic specifications. A static media description specifies all of the media for a job that the dynamic execution of the PostScript setpagedevice operator may request by page during job processing. Many applications that generate PostScript create static specifications, while only a small number of applications generate dynamic specifications.

The following sections describe these specifications in more detail.

Specifying static media

If media that matches a job's static media specification is loaded in a feeder tray and correctly specified to DocuPrint through a **Set Tray** command, the job prints using the media from the appropriate tray. If any of the static media requirements are not loaded, the job will be ineligible. The job does not print until you enter a **Set Tray** command indicating that media is loaded that matches the job's static media specification.

If MediaType, MediaColor, or MediaWeight are omitted from the static media specifications, they default to the Printer Controller Default Media values and the job prints using media from a tray that has been set (using the **Set Tray** command) to match the Default Media values.

If the Set Tray command is used to establish a feeder with no MediaType specified, only jobs with no type specified in the media attribute use that feeder tray. In addition, the MediaSize must be matched.

If MediaColor is also omitted from the Default Media, the MediaColor is set to "white".

If MediaWeight is set to its default of 0 in the Default Media, the job prints from a tray with any media weight specified.

If the Set Tray command is used to establish a feeder with MediaWeight=0 or no MediaColor, jobs with any color or weight specified in the media attribute can use that feeder tray providing the MediaSize and MediaType are matched.

Static media specifications can be set using a number of methods when the job is created or submitted, as described in the following sections.

Submitter

Submitter media specifications are specifications supplied by the submission protocol. They have the highest priority because they indicate the user preference at the time the job is submitted.

Two types of submitter methods are used by the protocol translators, affecting the use of Document Structuring Conventions (DSC) commands that specify media:

- When the job is submitted by the client, it requests values to be entered with the job. Subsequent DSC comments may supplement, but cannot override, these requests. For example, a Sun print client may request red paper for a job, and a DSC comment in the PostScript file may add that the job should print on USLegal paper.

  Refer to the *Guide to Submitting Jobs from the Client* for more information.

- A Job Ticket can be added to the front of a PostScript job before the job is sent to the DocuPrint. The Job Ticket specifies printing and media requests and supersedes the interpretation of DSC comments. For example, a document created on a Macintosh using the LaserWriter driver would use the **%%DocumentMedia** DSC comment for its static media specification, and **setpagedevice** to specify per-page media requests. However, if the document is saved on the Macintosh and then submitted using XPPS (the Macintosh submission client), the Job Ticket added by the submission client supersedes the DSC static media specification and the **setpagedevice** operators in the job may have no effect.

  Note: If there are external media specifications, the other three sources of static media specifications (Virtual Printer, DSC, and default) are ignored.

Submission Protocol

Job submission protocols such as lpr pass media attributes through the network protocol.

Note: lpr passes only attributes included in the -C field.

Virtual Printer

If there are no media specifications from the job submission client or the submission protocol, the media attribute assigned to the target virtual printer will take effect (unless SkipJTPrescan attribute is set on Virtual Printer).The following is an example of a multiple media value assigned to a virtual printer:

- Change Virtual Printer
- Enter virtual printer name: MailMerge

- Enter the attribute name: Media
- Enter the attribute value: USletter::white,USletter::blue, USletter:letterhead:white

Note: The virtual media printer attribute takes precedence over the **%%DocumentMedia** DSC command.

Document Structuring Conventions

If necessary, DocuPrint can use the PostScript Document Structuring Conventions (DSC) to determine static requests for document media. If there are no media specifications in the job submission (both the submission client and submission protocol) or media specifications assigned through the virtual printer, the DSC are parsed and the media specifications in the DSC become the media specifications for the job.

Note: The PostScript Policy Mechanism is not implemented.

The following is an example of a **%%DocumentMedia** DSC command:

```
%%DocumentMedia: plain 612 792 75 white logo
```

where*:*

Plain=tag name of the media. This is not used on DocuPrint NPS, but must be present in the comment for the document to exit the processing state. The actual name does not matter.

612 792=the paper size measured in PostScript units (72) per inch.

A4	72 x 8.26=595	72 x 11.69=842
USLetter	72 x 8.5=612	72 x 11=792
USLegal	72 x 8.5=612	72 x 14=1008
A3	72 x 11.69=842	72 x 16.54=1191

75=weight of the media in grams per square meter (20lb=75gsm, 65lb=176gsm). This number can be found on the side of a paper ream next to the Substance entry. This is a relative value and does not have to exactly match the weight of the paper itself but must match the attribute of the **Set Tray** command.

White=the media color

logo=the media type, such as the name of a customer form.

Always enter a value (even if the value is null) for the type, color, and weight when using the **%%DocumentMedia** command. The null value for numeric items is 0, and the null value for text items is ().

DSC can also be used to specify PageOrder and a subset of Requirements: simplex, duplex, numcopies, resolution, and staple. Any subset of these selections is allowed. The order of the comments is not important. The following example will print five copies, print on two sides, and staple each document:

```
%%Requirements: numcopies(5) duplex staple
```

Note: When printing duplex, the **setpagedevice** operator does not take effect until the next “front-side” of a sheet.

For more information about DSC, refer to the Adobe PostScript Language Reference Manual, 2nd edition.

Default

If there are no media specifications in the job submission, no media specifications in the virtual printer, and no media specifications in the DSC, the system-wide default media specification becomes the media specification associated with the job.

Specifying dynamic media

Dynamic media specifications are associated with pages in a job. This allows a job to print with different media on a per-page basis. The assigned media remains in effect until it is changed by another dynamic specification.

A subset of the features of the PostScript **setpagedevice** operator is supported for dynamic media requests:

- *MediaType*
- *MediaColor*
- *MediaWeight*
- NumCopies (must be specified before the first page is completed)

Note: Remember that the **setpagedevice** command executes the **initgraphics** and **erasepage** commands.

The precedence of media characteristics that are checked is as follows:

- *MediaSize*—preset in a job ticket, DSC, or set to the default media of printer
- *MediaType*—type of media, such as the name of a customer form
- *MediaColor*—paper color
- *MediaWeight*—grams per square meter; 20lb=75gsm. This number can be found on the side of a paper ream next to the Substance entry. This is a relative value and does not have to exactly match the weight of the paper itself but must match the attribute of the **Set Tray** command.

During job processing, the **setpagedevice** operator specifies a media for printing a specified page. However, it is not possible to specify a tray number for media selection. The interpretation of the **setpagedevice** operator dynamically determines the tray that the printer uses to feed the paper on which the corresponding page is printed. The selected tray contains media that best matches the media specified.

There are three media matching possibilities.

- Complete match—A tray with a media specification that matches the media specified in the **setpagedevice** operator. When there is more than one tray that is a complete match, one of the trays is arbitrarily chosen.

Note: The tolerance for media size match is ±2 mm (approximately 5 default user space coordinates).

- Partial match—A tray with a media specification where some fields match the media specified in the **setpagedevice** operator. If there are no complete matches, one of the partially matching trays is arbitrarily chosen.
- No match—None of the trays with media specifications match any of the fields in the **setpagedevice** operator. If there are no matches, a tray is arbitrarily chosen, provided that a tray contains media of a size that matches the static media specification.

Note: The **setpagedevice** operator will not signal the printer to stop printing if the selected media is not available. Only the **%%DocumentMedia** DSC command will mark the job ineligible if the needed media is not available.

You should always enter a value (even if the value is null) for the type, color, and weight when using the **setpagedevice** operator. The null value for numeric items is 0, and the null value for text items is ( ).

The following is an example of a **setpagedevice** operator used to set the current paper type to plain, white, and 75 gsm:

```
%%Page: "" 1
<</MediaType (plain)
    /MediaColor (white) /MediaWeight 75 >>
setpagedevice
%%BeginPageSetup
```

The following is an example of a **setpagedevice** operator used in conjunction with the **%%DocumentMedia** DSC command to ensure the printer waits for the proper media to be loaded before printing.

```
%%DocumentMedia: plain 612 792 75 white ()
%%+ slipsheet 612 792 75 blue ()
/Helvetica findfont 30 scalefont setfont
100 100 moveto
(this is page 1) show
showpage
%END OF FIRST PAGE
<</MediaType ()
    /MediaColor (blue) /MediaWeight 75 >>
setpagedevice
```

Mixed paper sizes

Mixed paper size within a job is supported on all NPS models. The number of stocks, sizes and types used is dependent on the number of trays available on the system.

Job specification

The paper sizes are specified in either the file generated by the application or via the ExceptionPage attribute. The different paper sizes may be in any pitch mode (or slow scan direction). Duplex is supported for all paper sizes.

Job performance

Use of the mixed paper size feature causes some degradation of speed. The maximum printing speed will be determined by the slowest paper size in the job (page buffers will be the size of the largest page). Where different pitch modes are used, performance is most affected. Orientation changes will further degrade performance. The addition of more memory should improve performance.

Tab printing on the 96/4635/180 NPS

The support for mixed paper sizes and ordered stock also allows the use of tabs. For example, tabs to be inserted in jobs whose standard stock is 8.5 x 11 inches typically have dimensions of 9 x 11 inches.

Note: Tabs are supported only on the side edges, not along the top or bottom.

For 4635 printers, the printing of duplex tabs is only possible on a model II 4635 printer. A field upgrade is available for model I IOTs to allow duplex tab printing.

True IOT plex changes within a job is not supported, however PostScript and PCL jobs can simulate this by generating blank pages.

Including tabs within a job

DocuPrint NPS can print tabs within a job whether or not the software application used to create the job is able to include mixed paper sizes for printing on the tab. If the application cannot mix normal and tab stock, then AutoTabShift must be used.

Set Tray and load media

At the Printer Controller, ensure that all stocks for the job, including the tab stock, are in the appropriate trays and that the Set Tray command has been used for each tray to define the type of stock in the tray.

Example

A media specification for 9 x 11 inch (229 x 279 mm) tabs that are precut with five items in an ordered set is:

```
229x279:precuttab-ordered5:white
```

If the requested media is not loaded, then the job will be ineligible.

Defining pages for tab stock

When submitting a job with mixed media that includes tabs, you can use Exception Pages or PostScript dynamic media selection to specify on which pages the tab stock should be used.

Exception Pages can be specified by:

- an application that generates a Xerox job ticket (such as XDS, a Xerox driver, the Xerox print command)
- Change Document Attributes command.

For more information on using Exception Pages when submitting your job, see the *Guide for Submitting Jobs from the Client*.

For PostScript jobs, dynamic media specifications within the PostScript code can request stocks on a page-by-page basis. (See the section “Specifying dynamic media” in this chapter.)

There must be a static media definition that specifies the media used in the job prior to processing the PostScript. An example of a static media definition is a virtual printer attribute.

Using AutoTabShift

The AutoTabShift attribute is intended to be used when the document requires a mix of normal and tab stock, and there is no support for printing on the tabs in the creation application.

When the AutoTabShift attribute is True, and a job has mixed media sizes, then for any selected medium that contains the text string "tab" in the type field of the media specification, the page image is shifted by the difference between the selected medium width and the default medium width for the job. (On the front side of the sheet the page contents are shifted to the right; on the back side of a duplex sheet, the contents are shifted to the left.)

Example

For example, consider a typical job where there are two media:

- The first medium (the default medium for the job) is usletter::white (or 216x279::white)
- The second medium is tab stock, specified as 229x279:precuttab-ordered5:white

Then for pages that have the tab stock selected and are on the front side, a right shift of 13 mm (0.5 inch) will result. This allows printing on the tabs.

Setting AutoTabShift attribute

The AutoTabShift attribute can be set on a virtual printer by using either the Change Document Attributes command or the Xerox Client software print command.

Decomposition Service forms in the Byte Compression format (BC) will not shift, so forms requested via the background form attribute don’t shift. If a form using the G4 format is called out explicitly from the job’s PostScript code, then the tab shift will take place. This is consistent with other horizontal shifts.

Ordered stock on the 96/4635/180 NPS

The Models 96, 4635, and 180 NPS have a feature which allows the printing of jobs containing ordered stocks. Ordered stocks consist of paper stacked in a feed tray in a cyclic fashion. For example, tray 1 may consist of colored paper ordered pink, green, blue, pink, green, blue and so forth.

Tab stocks is a special case of ordered stocks. This consists of tab sheets whose tab positions are arranged in cyclic order.

Ordered stock is specified as part of the media definition using the “type” field. Valid options for the type field include:

> [“transparency”] [“precut” | “drilled” | “drilled-precut”] [“#hole”]
> [“ordered#”]
>
> For #hole, the # symbol is replaced by a number indicating the number of holes drilled in the stock.
>
> For ordered#, the # symbol is replaced by a number indicating the number of items in the ordered stock set. The ordered stock option can have up to 255 items.

The Set Tray command has an attribute to specify the number of the item in the cycle that is positioned on the top of the tray. It is:

> **Set Tray <Tray#> Position <Position #>**

Note: The Set Tray Position command is not yet implemented. It is not required for ordered stock applications.

Checking media in the printer

Because DocuPrint NPS will hold print jobs until the correct media is loaded in the Printer Controller, you may want to check the media currently available in the printer before you send the job. To do so, use the **currentpagedevice** operator, which returns a dictionary reflecting the state of the printer when the job is running. This dictionary includes the following information:

- *PageSize*
- *MediaType*
- *MediaColor*
- *MediaWeight*
- *NumCopies*
- *Duplex*
- *Tumble*
- *Staple*
- *HWResolution*
- *InputAttributes*

Note: You can also use the Printer Controller Show Status command to determine current media.

5. Specifying multiset print jobs

The MultiSet feature in DocuPrint NPS is designed to allow multiple documents to be submitted as a single file or job while still treating them separately in the output by stapling, copying, or offsetting.

A job that does not use the MultiSet feature is treated as a single document. Only the entire job can be stapled or offset from other jobs, and only collated copies of the entire job can be requested. However, a job that uses the MultiSet feature can be divided into multiple documents. Each individual document can either be stapled or left unstapled, and collated copies can be requested for any of the documents. A document and copies of that document can also be offset from the other parts of the job.

A MultiSet job includes one or more sets, and each set includes one or more runs. An individual run can be stapled or left unstapled, but sets composed of multiple runs cannot be stapled. An individual set can be offset from other sets, but runs cannot be offset. Multiple copies can be requested for individual sets or runs, or both.

Sets and runs are identified by special entries in the **setpagedevice** command that must be present in the job:

- EndOfSet
- EndOfRun
- Staple
- Jog

Successful job construction requires the proper use of these commands in a job. The following sections describe the syntax and use of the **setpagedevice** command and its operators, and discuss the effects of these operators on decomposition.

setpagedevice

The MultiSet feature uses the PostScript **setpagedevice** operator to delineate sets and runs within a job, and to determine whether or not a particular set or run is offset, stapled, or copied. The MultiSet calls documented here provide an extension to the **setpagedevice** operator, and are governed by rules documented in the *PostScript Language Reference Manual*, 2nd edition, which include the following:

- Calls to **setpagedevice** place key-value pairs in a page device dictionary.
- Calls to **setpagedevice** are cumulative. Multiple calls on the same page of the document can change different key-value pairs within the page device dictionary without changing other values stored in the dictionary. However, all values in the dictionary can be changed if the device is restored to some

previous state using the **restore, grestore, grestoreall, or setgstate** operators.

- Any call to **setpagedevice** implicitly performs **erasepage** and **initgraphics**. Because the **erasepage** procedure is performed, any **setpagedevice** calls must precede the placement of lines, characters, and graphics on the page.
- PostScript programmers should be cautioned against overusing the **setpagedevice** operator because this will lead to a decrease in performance due to the additional time required for the **erasepage** operator.
- The **initgraphics** operator will not change the contents of the page device dictionary.
- There is no way to deactivate usage of the page device dictionary.

The following is an example of a MultiSet **setpagedevice** call:

```
<< /EndOfRun false /EndOfSet false >>
setpagedevice
```

The MultiSet calls provide an extension to the **setpagedevice** operator described in the Adobe specification. When using MultiSet, use caution when writing PostScript files for general PostScript devices. Use the **currentpagedevice** operator to determine if the MultiSet key-value pairs are available on the PostScript device in use. The following example uses **currentpagedevice** in a job that is not device specific:

```
currentpagedevice /EndOfSet known {

currentpagedevice /EndOfRun known {

<< /EndOfRun false /EndOfSet false >>
setpagedevice

        } if

    } if
```

For other information about use of the **setpagedevice** operator with DocuPrint, refer to the “Specifying Media” chapter.

EndOfSet

The EndOfSet command is used to indicate the beginning and end of a set. It can also be used to specify the number of copies of the set.

To signify the end of a set, the EndOfSet key must either be set true or set to a non-zero number on the final page of the set. Because the page device dictionary does not lose values set previously in the job, the EndOfSet key must be set to 0 or false on the first page of the next set. If EndOfSet is allowed to remain true, each subsequent page will be delivered to the output as a separate set.

Normally, sets containing multiple runs cannot be stapled. However, if a job contains only sets and no runs, each set will also be a logical run and will be stapled if the Staple attribute is set for the job.

Sets can be offset by setting the Jog value in the page device dictionary to 3. The Jog command is described later in this chapter.

PostScript Syntax << /EndOfSet <number | true | false> >> setpagedevice

Arguments <number>

If the number is 2 or greater, the current page is the last page of a set and <number> copies of the set are printed. A value of 1 results in the same behavior as a value of true. A value of 0 results in the same behavior as a value of false.

false

The current page is not the last page of the set.

true

The current page is the last page of the set.

The value of EndOfSet is false if it is not explicitly set in the job.

Example The following table illustrates an example of a job containing sets. For an example using both sets and runs, see “Using EndOfSet and EndOfRun in the same job.”

Table 5-1. **Job containing sets**

Stacker Output Sheet #: Set #, Page #	**Page # of job, setpagedevice command**
Sheet 1: Set 1, Page 1	Page 1, << /EndOfSet false>>
Sheet 2: Set 1, Page 2	Page 2, << /EndOfSet true >>
Sheet 3: Set 2, Page 1	Page 3, << /EndOfSet false >>
Sheet 4: Set 2, Page 2	Page 4, no setpagedevice command
Sheet 5: Set 2, Page 3	Page 5, << /EndOfSet 3 >>
Sheet 6: Set 2, Page 1	Page 3, << /EndOfSet false >>
Sheet 7: Set 2, Page 2	Page 4, no setpagedevice command
Sheet 8: Set 2, Page 3	Page 5, << /EndOfSet 3 >>
Sheet 9: Set 2, Page 1	Page 3, << /EndOfSet false >>
Sheet 10: Set 2, Page 2	Page 4, no setpagedevice command
Sheet 11: Set 2, Page 3	Page 5, << /EndOfSet 3 >>
Sheet 12: Set 3, Page 1	Page 6, << /EndOfSet false >>
Sheet 13: Set 3, Page 2	Page 7, no setpagedevice command
Sheet 14: Set 3, Page 3	Page 8, << /EndOfSet true >>

If the Staple attribute is set for the job in the example above, each set will be stapled, because no runs are specified within the sets. Use of runs for stapling is described in the next section.

If the Jog command is set to a value of 3 at the beginning of this example, each set will be offset from the other sets.

EndOfRun

The EndOfRun command is used to indicate the beginning and end of a run. It can also be used to specify the number of copies of the run.

To signify the end of a run, the EndOfRun key must either be set true or set to a non-zero number on the final page of the run. Because the page device dictionary does not lose values set previously in the job, the EndOfRun key must be set to 0 or false on the first page of the next run. If EndOfRun is allowed to remain true, each subsequent page will be delivered to the output as a separate run.

If a job contains only runs and no sets, the entire job will be treated as a single set and DocuPrint NPS will not begin delivering sheets until the entire job is decomposed.

Unlike sets, runs cannot be offset using a Jog command.

PostScript Syntax

<< /EndOfRun <number or true or false> >> setpagedevice

Arguments

<number>

If the number is 2 or greater, the current page is the last page of a run and <number> copies of the run are printed. A value of 1 is the same as a value of true. A value of 0 is the same as a value of false.

false

The current page is not the last page of the run.

true

The current page is the last page of the run.

The default value of EndOfRun is false if it is not explicitly set in the job.

Example

The following table illustrates an example of a job containing runs.

Table 5-2. **Job containing sets**

Stacker Output Sheet #: Run #, Page #	Page # of job, setpagedevice command
Sheet 1: Run 1, Page 1	Page 1, << /EndOfRun false>>
Sheet 2: Run 1, Page 2	Page 2, << /EndOfRun true >>
Sheet 3: Run 2, Page 1	Page 3, << /EndOfRun false >>
Sheet 4: Run 2, Page 2	Page 4, no setpagedevice command
Sheet 5: Run 2, Page 3	Page 5, << /EndOfRun 3 >>
Sheet 6: Run 2, Page 1	Page 3, << /EndOfRun false >>
Sheet 7: Run 2, Page 2	Page 4, no setpagedevice command
Sheet 8: Run 2, Page 3	Page 5, << /EndOfRun 3 >>
Sheet 9: Run 2, Page 1	Page 3, << /EndOfRun false >>
Sheet 10: Run 2, Page 2	Page 4, no setpagedevice command

Table 5-2. **Job containing sets** (continued)

Stacker Output Sheet #: Run #, Page #	Page # of job, setpagedevice command
Sheet 1: Run 1, Page 1	Page 1, << /EndOfRun false>>
Sheet 11: Run 2, Page 3	Page 5, << /EndOfRun 3 >>
Sheet 12: Run 3, Page 1	Page 6, << /EndOfRun false >>
Sheet 13: Run 3, Page 2	Page 7, no setpagedevice command
Sheet 14: Run 3, Page 3	Page 8, << /EndOfRun true >>

If the Staple document attribute was set for the job, all runs would be stapled in this example.

Using EndOfSet and EndOfRun in the same job

The recommended method for using the MultiSet feature is to use EndOfSet and EndOfRun together in the same job so that runs can be stapled, and a set containing multiple runs can be processed while a previous set is printing. In this way, the PostScript can make efficient use of the DocuPrint NPS Printer Controller CPU time to process a set while the operator handles the set currently delivered to the stacker. In addition, the sets can be offset from each other for easy removal from the output bin.

The following table illustrates an example of a job using the EndOfSet and EndOfRun commands in the same job.

Table 5-3. **Job using EndOfSet and EndOfRun**

Stacker Output Sheet #: Run #, Page #	Page # of job, setpagedevice command
Sheet 1: Set 1, Run 1, Page 1	Page 1, << /EndOfSet false /EndOfRun false >>
Sheet 2: Set 1, Run 1, Page 2	Page 2, << /EndOfSet false /EndOfRun true >>
Sheet 3: Set 1, Run 2, Page 1	Page 3, << /EndOfSet false /EndOfRun false >>
Sheet 4: Set 1, Run 2, Page 2	Page 4, no setpagedevice command
Sheet 5: Set 1, Run 2, Page 3	Page 5, << /EndOfSet false /EndOfRun 2 >>
Sheet 6: Set 1, Run 2, Page 1	Page 3, << /EndOfSet false /EndOfRun false >>
Sheet 7: Set 1, Run 2, Page 2	Page 4, no setpagedevice command
Sheet 8: Set 1, Run 2, Page 3	Page 5, << /EndOfSet false /EndOfRun 2 >>
Sheet 9: Set 1, Run 3, Page 1	Page 6, << /EndOfSet false /EndOfRun false >>
Sheet 10: Set 1, Run 3, Page 2	Page 7, << /EndOfSet true /EndOfRun false >>
Sheet 11: Set 2, Run 1, Page 1	Page 8, << /EndOfSet false /EndOfRun true >>
Sheet 12: Set 2, Run 2, Page 1	Page 9, << /EndOfSet false /EndOfRun false >>
Sheet 13: Set 2, Run 2, Page 2	Page 10, no setpagedevice command
Sheet 14: Set 2, Run 2, Page 3	Page 11, << /EndOfSet true /EndOfRun true >>

If the Staple document attribute was set for the job, all runs would be stapled in this example except for Set 2, Run 1, which is only 1 page.

Because MultiSet jobs are decomposed on a set-by-set basis, there may be some confusion about the order in which the output is delivered to the stacker. If the Staple attribute was set in the previous example, the job would be delivered to the stacker in the following order:

- Set 1, Run 3 face-up, on the bottom
- Set 1, Run 2, face-up, on top of Set 1, Run 3
- Set 1, Run 1, face-up, on top of Set 1, Run 2
- Set 2, Run 2, face-up, on top of Set 1, Run 1
- Set 2, Run 1, face-up, on top of Set 2, Run 2

Staple

The Staple command is used to indicate whether or not a run should be stapled. It can also be used to specify the end of a run. However, setting Staple does not change the value of EndOfRun in the page device dictionary.

Note: Since the High-Capacity Stitcher/Stacker requires documents to be delivered to the stitching unit face-up, documents to be stapled must be delivered in reverse page order, with the last page of the document delivered to the stitcher first. Because runs can be either stapled or not stapled but sets cannot, DocuPrint must convert an entire set of multiple runs into raster data (decomposition) prior to imaging sheets for delivery to the stitcher. If a set contains thousands of pages composed of hundreds of runs, decomposition will require a considerable amount of time. It is best to break such jobs into more manageable sets with a small number of runs in each set. The ideal number of runs to place in each set varies by the complexity of the PostScript. Although this is completely subjective, when using PostScript code of average complexity with runs composed of 10–25 pages, 5 runs would probably be a good number to print in each set.

In addition to setting the Staple value in the page device dictionary, the Staple document attribute must also be set for the job. If the Staple attribute is not explicitly set in the job, the value of the Staple key in the dictionary is assumed to be 0 by default. If the Staple document attribute is set for the job, the value of the key is assumed to be 1 by default.

The Staple attribute can also be set for the job using the PostScript Document Structuring Convention (DSC) comment "%%Requirements: staple" in the comments section at the beginning of the document. Or, it can be set using the print submission client, the Windows submission client, by submission to a virtual printer with the attribute set, or using the -C"(stitch)" option from lpr.

As with using the Run command by itself, if a job contains only Staple commands and does not use sets, the entire job will be treated as a single set with multiple runs and DocuPrint will not begin delivering sheets until the entire job is decomposed.

PostScript Syntax << /Staple <0 | 1 | 4> >> setpagedevice

Arguments 0

Do not staple the current run. Changing the value of Staple from 1 to 0 indicates the beginning of a new run starting with the current page.

1

Staple the current run. Changing the value of Staple from 0 to 1 indicates the beginning of a new run starting with the current page.

4

Indicates that the current page is the last page of the current run. If this value is used, the Staple key must set to 0 or 1 on the next page. If the value of Staple remains set to 4, each subsequent page will be treated as a separate run.

Example The following table illustrates an example of a job using the Staple command.

Table 5-4. **Job using Staple**

Stacker Output Sheet #: staple	**Page # of job, setpagedevice command**
Sheet 1, unstapled	Page 1, << /Staple 0 >>
Sheet 2, unstapled	Page 2, no setpagedevice command
Sheet 3, stapled to Sheet 4	Page 3, << /Staple 1 >>
Sheet 4, stapled to Sheet 3	Page 4, << /Staple 4 >>
Sheet 5, stapled to Sheet 6	Page 5, << /Staple 1 >>
Sheet 6, stapled to Sheet 5	Page 6, no setpagedevice command
Sheet 7, unstapled	Page 7, <</Staple 0 >>
Sheet 8, unstapled	Page 8, no setpagedevice command

Using the EarlyStaple option with MultiSet jobs

Documents requesting stitching must be entirely decomposed prior to imaging. This leads to long decomposition times if you set the Staple attribute on all submitted jobs, including jobs composed of thousands of pages. The EarlyStaple option was added so that an entire document composed of thousands of pages is not decomposed prior to imaging. When the EarlyStaple option is set to true, DocuPrint first decomposes only the first 50 sheets of the document, which is the physical limitation of the high-capacity stitcher/stacker. If the PostScript interpreter determines the document exceeds 50 sheets of paper, the document will be delivered to the stacker face-down, as if Stapling were not set for the job.

With the MultiSet feature, however, jobs are decomposed on a set-by-set basis, and each set can be composed of multiple runs, with each run requiring a separate staple. If only one of the runs in a set exceeds 50 sheets, it would not make sense to deliver all of the runs in the set face-down as if Stapling were not set for the job. So if EarlyStaple is set to true, and a run requiring more than 50 sheets is encountered within a set, a logical end of the set is forced at the end of that run. All runs encountered in the set up to the run of more than 50 sheets are delivered as one set and all runs after the run exceeding 50 sheets are delivered as a second set.

For example, if EarlyStaple were true and Set 1, Run 2 in the previous example exceeded 50 sheets, the output in that example would be delivered to the stacker in this way:

- Set 1, Run 1, face-up, on the bottom (logical end of Set)
- Set 1, Run 2, face-down, on top of Set 1, Run 1
- Set 1, Run 3, face-up, on top of Set 1, Run 2
- Set 2, Run 2, face-up, on top of Set 1, Run 3
- Set 2, Run 1, face-up, on top of Set 2, Run 2

Because of this confusing output, it is recommended that you use EarlyStaple if you use only sets in your jobs.

Jog

Use the Jog command to offset sets within a job or to add offsets to a job without delineating it into sets.

Unlike the Staple command, the Jog command has no effect on when a set or run ends. The value of the OffsetPerCopy and OffsetPerJob options have no effect on the Jog command.

Postscript Syntax <</Jog <0 | 1 | 3> >>setpagedevice

Arguments 0

Do not offset sets or sheets within this job.

1

Offset the next sheet of paper from the current sheet. The current value of EndOfSet has no effect on offsetting if this value is used. If the job is printed simplex, and Jog is set to 1, every sheet delivered to the stacker will be offset from the previous sheet until the Jog value is changed to 0 or 3. If the job is printed duplex or tumble duplex, the next sheet is offset from the current sheet when the Jog value is changed from 1 to 0 on either side of the current sheet.

3

Offset the next set from the current set. This value must be used in conjunction with the EndOfSet command to induce paper offsetting. This value will not cause the current set to be offset from any previous sets already delivered to the stacker.

The default value of Jog is 0 if it is not explicitly set in the job.

Specifying multiple copies

As seen in the previous sections, multiple copies of either sets or runs can be requested using the EndOfSet or EndOfRun commands by setting <number> to a value of 2 or greater.

The Copies document attribute can also be used to make copies of the entire MultiSet job. The Copies attribute has no effect on the copies of sets and runs requested within the job. Each copy of the entire job will contain multiple copies of any sets or runs requested in the page device.

For versions prior to 7.0, multiple copies of runs, sets, or jobs are always collated. Requesting uncollated copies is not an option.

For version 7.0 and later, multiple copies of runs, sets, or jobs are collated according to the value specified by the Uncollated attribute.

Handling duplexing in MultiSet jobs

When the document attribute for a MultiSet job is duplex or tumbleduplex, the EndOfSet, EndOfRun, and Staple MultiSet commands take effect immediately on the page whether that page is on the front-side or back-side of a sheet. The Jog command when used without EndOfSet, does not affect the output of a duplexed job until the back-side of the current sheet has been imaged.

The MultiSet feature does not allow both simplex and duplex in a job. You can create the appearance of mixed-plex by setting the Plex attribute to duplex for the job, and creating single page runs or sets using an EndOfRun, EndOfSet, or Staple command, thus forcing a new sheet of paper to be pulled for every single-page run or set. The disadvantage to the approach is that the back-sides of the sheets of paper are still imaged in the print engine and count as pages on the billing meters.

The following is an example of a duplexed MultiSet job.

Table 5-5. **Job using Duplex**

Stacker Output **Sheet #, side: Run #, Page #**	**Page # of job, setpagedevice command**
Sheet 1, front: Set 1, Run 1, Page 1	Page 1, << /EndOfSet false /EndOfRun false >>
Sheet 1, back: Set 1, Run 1, Page 2	Page 2, << /EndOfSet false /EndOfRun true >>
Sheet 2, front: Set 1, Run 2, Page 1	Page 3, << /EndOfSet false /EndOfRun false >>
Sheet 2, back: Set1, Run 2, Page 2	Page 4, no setpagedevice command
Sheet 3, front: Set1, Run 2, Page 3	Page 5, << /EndOfSet false /EndOfRun 2 >>
Sheet 3, back: blank	
Sheet 4, front: Set 1, Run 2, Page 1	Page 3, << /EndOfSet false /EndOfRun false >>

Table 5-5. **Job using Duplex**

Stacker Output Sheet #, side: Run #, Page #	**Page # of job, setpagedevice command**
Sheet 4, back: Set 1, Run 2, Page 2	Page 4, no setpagedevice command
Sheet 5, front: Set 1, Run 2, Page 3	Page 5, << /EndOfSet false /EndOfRun 2 >>
Sheet 5, back: blank	
Sheet 6, front: Set 1, Run 3, Page 1	Page 6, << /EndOfSet false /EndOfRun false >>
Sheet 6, back: Set 1, Run 3, Page 2	Page 7, << /EndOfSet true /EndOfRun false >>
Sheet 7, front: Set 2, Run 1, Page 1	Page 8, << /EndOfSet false /EndOfRun true >>
Sheet 7, back: blank	
Sheet 8, front: Set 2, Run 2, Page 1	Page 9, << /EndOfSet false /EndOfRun false >>
Sheet 8, back: Set 2, Run 2, Page 2	Page 10, no setpagedevice command
Sheet 9, front: Set 2, Run 2, Page 3	Page 11, << /EndOfSet true /EndOfRun true >>
Sheet 9, back: blank	

Minimizing decomposition time

Although it is strongly recommended that PostScript programmers using the MultiSet feature use both sets and runs in their jobs to achieve maximum efficiency with a minimum amount of decomposition time, usually for reasons of simplicity, most programmers write applications that use only sets or only runs in their jobs. Jobs containing only runs are not recommended, because they tend to require long decomposition times prior to imaging pages on the printer. Another disadvantage is that runs cannot be offset from other runs. If a job contains only sets, however, each set can be offset from other sets, and each set can be treated as a logical run and stapled.

If you experience long decomposition times associated with a job using the MultiSet feature, the job should be examined to see how many runs are contained within a set. If the job contains only runs, it should be altered so that only sets are used in the job, and each set becomes a logical single run. If this still does not reduce the decomposition time because of extremely long sets, the EarlyStaple option can be used.

Example

The following figure shows an example of a PostScript file using MultiSet features.

```
%!PS-Adobe-3.0
%%Title: TestMultiSet.ps
%%Requirements: staple numcopies(1)
%%DocumentMedia: dmWhite 612 792 75 white ()
%%+ dmOther 612 792 75 white (other)
%%PageOrder: Ascend

/HF /Helvetica findfont 12 scalefont def
HF setfont

/yStep 14 def
/pgNum 1 def
/pgH 11 72 mul def
/pgW 8.5 72 mul def
/dx 10 def
/fillPage {
/ps exch def
/cy pgH 36 sub def
/sw ps stringwidth pop def
{
36 cy moveto
pgW 72 sub sw dx add div floor cvi { ps show dx 0 rmoveto } repeat
cy 48 lt {exit} if
/cy cy yStep sub def
} loop
/pgNum pgNum 1 add def
} bind def

/doPage { fillPage showpage } bind def

<< /Jog 3 /Staple 0 /MediaType (other) /EndOfSet false >>
setpagedevice
(set 1, run 1, pg 1) doPage

<< /Staple 1 /MediaType () /EndOfSet false >> setpagedevice
(set 1, run 2, pg 1) doPage
(set 1, run 2, pg 2) doPage

<< /Staple 4 /MediaType () /EndOfSet false >> setpagedevice
(set 1, run 2, pg 3) doPage


<< /Staple 1 /MediaType () /EndOfSet false >> setpagedevice
(set 1, run 3, pg 1) doPage
(set 1, run 3, pg 2) doPage

<< /Staple 1 /MediaType () /EndOfSet true >> setpagedevice
(set 1, run 3, pg 3) doPage
```

```
<< /Staple 0 /MediaType (other) /EndOfSet false >> setpagedevice
(set 2, run 1, pg 1) doPage

<< /Staple 1 /MediaType () /EndOfSet false >> setpagedevice
(set 2, run 2, pg 1) doPage
(set 2, run 2, pg 2) doPage
```

6. Creating print jobs with referenced content

Using the PostScript **file** operator, DocuPrint NPS enables you to print jobs that are composed of more than one PostScript master. For example, jobs may include various images that are stored separately because it is inconvenient or inefficient to combine these images into the PostScript master at print time. These separate data files can be PostScript documents or other types of readable files, such as images. When these files are made available to DocuPrint, they can be referenced individually from the PostScript master and thus are called “referenced content.”

Note: Use of the referencing capability is restricted to files that are local to the DocuPrint file system. Files that reside in file systems on other network nodes cannot be referenced for reason of performance, system security, and system integrity.

This chapter explains how to name referenced content files, make them available to DocuPrint, and reference them within files.

Note: For information on Background Forms available with the Decomposition Service that allows you to merge any document on an electronically stored background form, see the *Decomposition Service and Tools Guide* for more details.

Naming referenced content files

DocuPrint NPS uses a relative addressing scheme in conjunction with the PostScript file operator. This means that referenced files cannot be placed in any arbitrary directory on the Printer Controller, but must be in directories in a specific path. This simplifies the references that must be included in the master, enhances security, and makes it easier to maintain the PostScript files.

All referenced files must be placed in the DocuPrint file structure proceeding from the /var/spool/data path. DocuPrint constructs pathnames to the referenced content files beginning with this directory so that all referenced files are named relative to this path.

Referenced content files may be associated with a single PostScript job or they may be shared resources used by many jobs; for example, scanned signatures for corporate executives that are referenced from many jobs. A different naming convention is recommended for each type.

Naming single-use content files

Use the following convention to place and name single-use referenced content files:

<appName>/<jobName>/<fileName>

where <appName> is the name of the submitting application, <jobName> is a name used for the print job, and <fileName> is the name of the file.

Remember, the <appName>/<jobName> directory is relative to /var/spool/data. For example, for a job that is made up of a collection letter and a referenced content file with an image of a bad check, the full pathname for the reference file would be:

/var/spool/data/collections/job521/chk22456

This placement simplifies the management of referenced content files and enables automatic deletion. See “Removing referenced files from the DocuPrint system” for more information.

Naming shared resource content files

Place shared resource content files in directories that are named similarly to single-use files relative to /var/spool/data, but without a job name:

<appName>/<fileName>

For each application that places referenced files on DocuPrint, a unique <appName> must be generated and used in the path name. This subdirectory is required, and it should be the joint responsibility of site administration and the application development organization to ensure that <appName> is unique within the /var/spool/data directory.

Caution: Using the <appName> subdirectory is essential for proper maintenance of your DocuPrint system. You may incur additional maintenance costs beyond those in your maintenance or service contract if you do not maintain this directory. ⊘

Transferring referenced content files to the DocuPrint file system

Before you submit a PostScript file to DocuPrint NPS as a print job, make sure all referenced files are resident on the DocuPrint Printer Controller. Use Network File Services (NFS) filing operations to place files in the local DocuPrint file system. Using NFS is the only way that non-font supplemental files can be made available to PostScript programs executing on DocuPrint.

Note: See *Guide to Configuring and Managing the System* for information about configuring NFS for DocuPrint.

You can use NFS to make the DocuPrint /var/spool/data directory known and accessible to a networked workstation. To do so, you must export the /var/spool/data directory on the DocuPrint Printer

Controller so that applications can store files for later referencing. Similarly, the machine on the other side of the file transfer must import the DocuPrint file system. Because DocuPrint uses a relative addressing scheme with the PostScript **file** operator to maintain system integrity, you cannot use a directory structure other than /var/spool/data.

Exporting the DocuPrint directory

Export the /var/spool/data directory on the DocuPrint by creating a file in /etc called exports that contains the following line before you issue the export command:

```
/var/spool/data
```

Caution: To protect the integrity of the system, do not export higher subdirectories. Exporting other parts of the DocuPrint file system may endanger DocuPrint security and integrity. In addition, if copyrighted material in the DocuPrint file system is exported, you could be in violation of copyright protection laws. ⦸

If remote workstations are running UNIX, creating the /etc/exports file is the only step that you need to perform on the Printer Controller. If there are other operating systems in use, you may need to perform other steps on the Printer Controller to export the directory. For general information about installing non-UNIX NFS software, follow the instructions for the networking software, and read the section “Using an alternate NFS daemon” in the chapter “Handling XDOD/DigiPath jobs” in the *Guide to Configuring and Managing the System*.

In addition to exporting the /var/spool/data directory, you must create the following unique subdirectory for each application that places supplemental files on the Printer Controller.

Construct this setup process as a UNIX script and back it up so it can be used after an upgrade:

```
/var/spool/data/<appName>
```

Note: If there is more than one application, export the appropriate subdirectory for each application.

Importing the DocuPrint directory

If you have a remote workstation that must have NFS access to the DocuPrint /var/spool/data directory, you must import that directory and assign it to a mount point in the remote workstation file system.

If you are using a system other than UNIX that offers NFS capabilities, follow the manufacturer’s instructions for mounting the directory. If you are using a UNIX system, complete the following steps to mount the directory:

1. As root, create a directory to be used as the mount point.
2. Enter your DocuPrint system host name and ip address into the /etc/hosts table, if it is not already there or if you are not running DNS (Domain Name System).

3. Use the following command to mount the /var/spool/data directory.

   mount <hostname>:/var/spool/data /mydir

   <hostname> is the assigned hostname of the printer. The reference to mydir acts as a local directory.

   This command locates all referenced files on DocuPrint in the <appName> subdirectory of the /var/spool/data directory. You must create the <appName> directories in DocuPrint; you cannot use NFS to create them. See “Naming referenced content files” for information on naming conventions.

4. Transfer the files as needed. Use the following commands:

   - **ls /mydi**r lists files
   - **cp** copies files to or from the DocuPrint
   - **df** shows the DocuPrint /var/spool/data directory mounted as a file system on the workstation.

   When transferring files to the DocuPrint file system, use the following path to store them on DocuPrint:

   /<mountPoint>/<appName>/<jobName>/<fileName>

Note: If you need to disable NFS file transfer to DocuPrint at any time, enter the following command as root:

umount /mydir

Do not mount remote file systems from the Printer Controller because this will result in performance problems.

Referencing files within PostScript masters

To reference a file from within a PostScript master, create a line with one of the following formats:

```
(<appname>/<jobname>/myDataFile) (r) file
operatorseq
```

or

```
(<appname>/myDataFile) (r) file operatorseq
```

The file name and relative path are enclosed in parentheses (), the (r) string informs the PostScript interpreter that it should have read access, and the file operator identifies the preceding object as a file. The operatorseq string is a sequence of one or more PostScript operators that act on the file object.

The following examples show fragments of PostScript code that accomplish various actions with the file reference.

Figure 6-1. **PostScript fragment: using the file operator with images**

```
...
256 256 8 [256 0 0 -256 0 256]
(myDirectory1/myDirectory2/myDataFile) (r) file
/ASCII85Decode filter
/RunLengthDecode filter
image
...
```

Figure 6-2. **PostScript fragment: using the file operator with images**

```
...
/myName (myDirectory1/myDirectory2/myDataFile) (r) file def
myName 246 setfileposition
256 256 8 [256 0 0 -256 0 256]
myName
/ASCII85Decode filter
/RunLengthDecode filter
image
myName closefile
...
```

Figure 6-3. **PostScript fragment: using the file operator with EPS**

```
...
(myDirectory/myEPSFile) (r) file cvx exec
...
```

Referencing files using the GetTiff macro

The GetTiff macro enables a PostScript program to read a file in TIFF-compressed format and image it, possibly overlaying variable data. The macro is supplied with the DocuPrint Printer Controller, and simplifies referencing TIFF images. The PostScript necessary to decode the TIFF header is contained in the macro.

The GetTiff PostScript macro is designed to efficiently parse, decode, and image the TIFF, which can be uncompressed or compressed with CCITT Group 4 fax standard compression. The macro images, but does not print, the TIFFs, allowing many TIFFs to be placed over each other with no image data loss. This is because the PostScript imagemask operator is used; if the pixel in the TIFF contains white data, it is treated as transparent.

The GetTiff macro is typically used for printing imported TIFF images. GetTiff takes the name of the TIFF file and images it, as in the following example:

(Test.tif) GetTiff

This example code images the page stored in the file named Test.tif using the current matrix. You may need to include PostScript commands before and after the **GetTiff** command to save and set the graphics state, and to restore it after imaging the TIFF. You must store the files relative to the /var/spool/data directory. **GetTiff** uses the relative path conventions described in “Referencing files within PostScript masters.”

GetTiff is sensitive to the TIFF tags and parses the TIFF header to reorient the page if needed and to comprehend the setting of the compression tag (Uncompressed or G4).

In the current DocuPrint release, the GetTiff macro is already installed in the printer startup file; no additional installation or setup is necessary.

GetTiff capabilities

GetTiff users should be aware of the following capabilities:

- Only the following TIFF tags are used: IMAGEWIDTH, IMAGELENGTH,‘ BITSPERSAMPLE, COMPRESSION, PHOTOMETRIC, FILLORDER, STRIPOFFSETS, ORIENTATION, ROWSPERSTRIP, XRESOLUTION, YRESOLUTION, GROUP3OPTIONS, GROUP4OPTIONS
- All valid orientations are supported.
- Grayscale for raw data is supported.
- Multi-strip TIFF files are supported.
- TIFF files with bad compressed data issue warnings instead of errors.
- If ORIENTATION is missing or less than 1 or more than 8, then the value will be assumed to be 6 if the length is greater than the width, and 8 if the length is less than the width.
- If BITSPERSAMPLE is missing or less than 1, then the value will be assumed to be 1.
- If ROWSPERSTRIP is missing or is less than 1, then it will be assumed to be the total number of rows (true for a single strip file).
- All GROUP3OPTIONS bits (2DENCODING, UNCOMPRESSED, FILLBITS) are supported.
- The GROUP4OPTIONS value is ignored, but an attempt is made to support uncompressed runs of data within the compressed data according to the TIFF v. 6.x specification.
- Tags with types less than 1 or more than 5 are ignored.
- The imaging uses the imagemask operator for 1 bit/pixel data and the image operator for all other data. Setting UseImageMask to false in the GetTiffDictionary will force GetTiff to always use the image operator.
- FILLORDER is ignored for uncompressed data.

- XRESOLUTION and YRESOLUTION are required, must be of type rational, and are always assumed to be in samples/inch.

GetTiff limitations

Users should be aware of the following limitations.

- Only compression types 1 (raw uncompressed data), 3 (CCITT group 3) and 4 (CCITT group 4) are supported.
- Only single page TIFF files are supported.
- For compressed data, only binary data is allowed.
- For raw data, only 1, 2, 4, or 8 bit/pixel is supported.
- There is no support for color.

Removing referenced files from the DocuPrint system

Once files are transferred to a DocuPrint Printer Controller, they remain in the system unless someone removes them for cleanup as part of your site management procedures or as part of job processing.

Removing referenced files associated with a single print job

For files that are referenced by only a single job, use the Xerox programmatic mechanism to remove them from the DocuPrint Printer Controller. To use this mechanism, place the following statements in the job's PostScript file before all other PostScript statements:

%%XEROX-XIPP: Filename
<appName>/<jobName>/<fileName>

where <appName> and <jobName> are the lowest two subdirectories in the path with contents to be deleted, and <fileName> is the name of any one of the files in that path.

Caution: Statements in the PostScript file in the %%XEROX-XIPP section should only refer to files that are located in the subdirectory that was created as described in "Importing the DocuPrint directory." You could incur additional service cost beyond those in your maintenance or service agreements by not following these precautions. 🛇

Place one statement in the above format into the PostScript file for each referenced file in the job. These statements place all of the files under the control of the JPM.

Caution: If you use this method for deleting the content of a job, you should use it for all the referenced components of the job. 🛇

The following example shows the cleanup of referenced files using the %%XEROX-XIPP document structuring conventions in a PostScript file.

Figure 6-4. **Using XEROX-XIPP**

```
%!PS-Adobe-3.0
%%Creator: IsPrint
%%Title:
%%CreationDate:
%%XEROX-XIPP: Filename myApp/myJob/myFile2.tif
. . .
%%EOF
. . .
```

Removing referenced files associated with multiple print jobs

When files containing data such as forms or logos are referenced by many jobs, they may reside on DocuPrint indefinitely. If these files are to be maintained manually, you should create site management procedures, which include provisions to delete the files when you no longer need them and to reinstall them if an error occurs or they are inadvertently deleted.

Rather than maintain the files manually, you may want to create a workstation-based utility to transfer and maintain the referenced files that need to reside on DocuPrint for significant periods of time.

7. Mixed plex applications

As of release 7.x, DocuPrint supports true mixed plex applications for both PostScript and PCL jobs. Release 1.6.x (and prior software releases) provides support for simulated mixed plex applications.

This chapter provides you with the background information you need regarding your mixed plex applications.

Note: The DocuPrint NPS implementation of true mixed plex is not the same as the implementation for DocuSP systems. Refer to the DocuSP documentation for details.

Introduction to mixed plex

Many applications require that some pages of a job print one-sided (simplex), while other pages of the same job print either two-sided (duplex) or head-to-toe (tumbleDuplex). In releases prior to 7.x, this is achieved by means of "simulated mixed plex", a technique that requires a programmer to code the job so that padding pages are printed on the back of a duplex page to give the appearance of simplex printing. In the true mixed plex capability available with release 7.x, DocuPrint NPS attributes can be used to control plex changes in the document, and one-sided pages are treated as simplex.

Terminology

The following are some terms that are useful in understanding the discussion of mixed plex.

- Plex changes within the job’s code are referred to as **dynamic** plex changes.
- Plex specifications set by a DocuPrint NPS attribute are referred to as **static**.
- A **padding page** is a blank page image that forces the next page to appear in a particular location-either the front or back side of a sheet. For example, a PostScript page device key or PCL page boundary command can force the creation of a padding page.

See the *Guide to Using Page Description Languages* for information on the DocuPrint NPS support for PCL and PostScript.

Simulated mixed plex (for release 1.6x and prior)

To create a simulated mixed plex application in release 1.6x (and earlier releases), dynamic plex changes are included in the job's PostScript or PCL code. In addition, the DocuPrint NPS plex attribute for the job has to be set to duplex.

In this case, the use of duplex for the value of the plex attribute has two effects: it behaves as an initial plex value, and it has the additional effect of allowing dynamic plex changes within the job's code to take effect. If the plex value is set to simplex or tumble duplex, that value overrides any plex changes within the job.

One drawback to the use of simulated mixed plex is that a padding page is counted and charged as a page. In addition, the duplex paper path is used for all pages, even those that have the appearance of being simplex, or one-sided. Thus a job that has only a few pages printed on both sides still prints at the speed of a duplex job, where pages are delivered to the stacker at approximately half the speed of a simplex job.

True mixed plex (for release 7.x)

In the true mixed plex capability supported by release 7.x, dynamic plex changes within the job can be used, but you can also use NPS document attributes to achieve the plex changes.

Moreover, pages that are intended to be one-sided are printed and counted as simplex, while two-sided pages are printed and counted as duplex pages.

One important change with the implementation of true mixed plex for release 7.x is that simulated mixed plex is not supported. This means that the duplex value for the plex attribute does not have the same function it did earlier; this function has been replaced by the attribute PlexChangesDynamic, discussed in the section "Specifying mixed plex in release 7.x." When this attribute is used, jobs that previously used simulated mixed plex continue to print with no outward changes to the plex behavior. The only difference is in how the printer tracks the changes, and in the accounting.

Note: For the 4050/4090, 4850/4890, and DP92C NPS, the systerm must clear the paper path before each plex change. (The 96/4635/180 NPS does not do this.) Performance is affected to some extent for all printers at each plex change. The degree to which performance is affected depends on whether or not the paper path is cleared and on the number of plex changes within the job.

Specifying mixed plex in release 7.x

With release 7.x, you can specify plex changes in several ways:

- Dynamic—Within the job's PostScript or PCL code
- Static—By using NPS attributes
- By using both of the above methods, though this is generally not recommended.

Applications that include dynamic plex changes

The NPS attribute PlexChangesDynamic is used to specify whether you want to use the dynamic plex specifications in the job's code.

- If you want to use the plex changes within the job's code, set this attribute to True. The value you use for the plex attribute will be the initial plex, and the plex changes within the job will take effect.
- If you want to ignore the plex changes within the job's code, set this attribute to False. In this case, the value you use for the plex attribute will override any plex changes in the job's code.

Note: For jobs that previously used simulated mixed plex, set the PlexChangesDynamic attribute value to True, and the job will print as expected. It is not necessary to set the value of the plex attribute to duplex unless that is required as the initial plex for the application.

Using static plex changes in an application

You can use the ExceptionPages attribute to specify plex changes on a per-page basis. See the *Guide to Managing Print Jobs* chapter "Document attributes" for the complete syntax for this and all other attributes. In general, the syntax is:

```
ExceptionPages <startpage, endpage,> plex,
<plex value>
```

The following example shows that the Change Document Attribute command is used to make the following plex changes to document 123: pages 1 and 2 will print tumbleDuplex, pages 3 and 4 will print simplex, and pages 5 through 23 will print duplex. A total of 13 sheets will print and 23 page images are counted.

```
PS-Admin> Change Document Attribute 123
ExceptionPages

[1,2,plex,tumbleDuplex],[3,4,plex,simplex],

[5,23, plex, duplex]
```

If you set the attribute PlexChangesDynamic value to False, then only the value for the plex attribute and the plex changes specified using the ExceptionPages attribute will take effect.

If you set the attribute PlexChangesDynamic value to True, then there are several factors that can influence plex changes:

- The plex attribute value
- The dynamic plex changes in the job's code
- The ExceptionPages attribute value, which overrides any other plex changes for the page.

In general, it is recommended that you do not use both dynamic and static plex changes, as there may be unexpected results.

Rules of precedence for plex changes

Because there are several ways to specify plex changes for the same page, there are rules of precedence that govern which change will take effect.

- If there is no other plex specification for the page, the value for the NPS plex attribute is used.
- If there is a plex change specified in the job's code (and no ExceptionPages attribute for plex), that plex change takes effect if PlexChangesDynamic is set to True.
- If there is an applicable ExceptionPages value for the page, that value is always used (regardless of the value used for PlexChangesDynamic).

How pages are counted

One of the differences in behavior between release 7.x and prior releases is the method used for counting pages. In general, for pre-7.x simulated mixed plex jobs, each output sheet counts as two pages, while for 7.x true mixed plex, simplex output sheets count as one page, and duplex sheets count as two pages.

There are some applications where other factors influence how pages are counted by the system, these are discussed in the next sections.

Page numbering prior to release 7.x

For releases prior to 7.x, DocuPrint NPS counts pages in the same way for PostScript and PCL. There are three cases:

- If the job is 1-sided, each output sheet counts as a single page.
- If the job uses simulated mixed plex, each output sheet counts as two pages, regardless of whether a particular sheet is 1-sided or 2-sided.
- If the job is simply 2-sided, the situation is slightly more complicated for PostScript. For PCL, each output sheet counts as two pages. For PostScript, each output sheet counts as two pages except in one case. If a /EndOfSet or /EndOfRun key introduces a padding page, that page is not counted. In that situation, the sheet with a padding page counts as one page; all other sheets count as two pages.

Page numbering for multiset applications

If a job uses multiset and specifies more than one copy for a run or a set, NPS only counts the pages in that run or set once, not once for each copy. For example, consider a 1-sided job with pages ABCDEFG where pages C, D, and E are in a separate run for which four copies are requested. For page numbering purposes, this job has seven pages, although sixteen pages are printed. Now assume the user requests a media exception on the fourth page. In this case, the exception is applied to all four copies of page D.

Other page numbering considerations

Users can request that only a subset of a job's pages be printed. Additionally, they can request that the job's pages be reversed. While doing either of these things does not explicitly provide plex information, it is important to understand how these page numbers interact with page numbers used in other places.

Page rages

Essentially, when a job requests page range N-M, the system produces pages 1-M taking into account all other requests, including plex changes and media exceptions. It then discards the first N-1 pages and outputs the rest, always starting on the front side of a sheet.

There are several consequences of this:

- All plex and media changes requested before page N take effect. So, for example, if the request is to print pages 5-10 and there is a plex change from 2-sided to 1-sided on page 4,then page 5 is printed 1-sided (unless, of course, page 5 requests otherwise).
- If the specified page range would have started on the front side of a sheet if all pages were printed, then the output will appear exactly as it would in that case.
- If the specified page range would have started on the back side of a sheet, the output will be different. Page ranges always start on the front side of a sheet. In this situation, all output pages are pushed to the opposite side; that is, backs become fronts and fronts become backs. This can cause image-shift problems.

In general, it is recommended that you specify page ranges that normally start on the front side of a sheet.

Descending page order

When the user requests a job be printed in descending page order, that request can interact in perhaps unanticipated ways with other requests, such as media exceptions.

Essentially, as with getting a page range, reversing the page order happens last. For example, assume you have a job with pages ABCD and request to print the job in descending page order, with the media exceptions on the third page. The system prints the job as DCBA, with the media exception applied to page C.

Page numbering for release 7.x

In general, for purposes of applying exceptions (for example, plex changes), pages are numbered as they would appear if the job were printed with a simplex override. That is, blank sides caused by plex changes are not counted.

The remainder of this section provides more detailed information on page numbering.

Note: This section discusses how pages are numbered for plex, media, and image shift exceptions. A different numbering scheme is relevant for accounting. Page numbering for exceptions is concerned with the job as it enters for decomposition. Page numbering for accounting is concerned with the pages as they reach the printer.

Within a data stream, page numbers are irrelevant. Commands in the data stream affect the current and subsequent pages, whatever their numbers may be. Page numbers become relevant when information is specified external to the data stream (for example, when using ExceptionPages). It is very important that however pages are numbered, a page number always refers to the same page image. For example, if page number 234 has a plex exception and page number 234 has a media exception, the same physical page on the same side of the same sheet should be affected by both of these exceptions.

DocuPrint NPS 7.x uses a numbering scheme based on the underlying data stream. This numbering scheme is designed to not count padding pages at all. In short, only page images created by the decomposers are counted.

- For PostScript, the page count increments by one for every showpage or copypage and does not increment for any other reason.
- For PCL, the system counts the page images created by the decomposer. It will count "normal" pages, but will not count padding pages created by any of the PCL plex commands.

A good guideline is that page numbers correspond to the pages that would be printed if the plex attribute value were simplex, and the PlexChangesDynamic attribute were false, and no plex changes are specified using the ExceptionPages attribute.

Because PostScript is a full programming language, it will always be possible for a particular PostScript application to exhibit unanticipated behavior. For example, the PostScript job might change the page content based on the page size, potentially causing extra showpages in the face of media exceptions. If so, the page numbers would change. This is a very unusual case.

Other page numbering considerations

Multiset

If a job uses multiset and a run or set has a copy count larger than 1, only the first copy is counted. Also, if an exception is specified for a page within a run or set that has multiple copies, that exception will apply to all copies of the page. This is consistent with pre-7.x NPS behavior.

Page ranges and descending page order

The behavior is the same as the pre-7.x NPS behavior described earlier.

Page numbering for accounting

For purposes of accounting, pages are counted as for exception page numbering, but the number of copies of each page is also included (for example, two copies of three pages counts as six pages, with the last page being page number 3). In addition to page counting, simplex and duplex sheets are accounted for separately.

Side selection for PCL mixed plex applications

When pages are printed in a two-sided mode, some PCL commands force the page following the command to break to a specific side. The PCL commands that force the page break are: Duplex Page Side Selection, Simplex/Duplex Print, Paper Source, Page Size, and Number of Copies. Pages containing these commands are always positioned as defined here, regardless of whether the command is redundant (for example, a Paper Source command selecting the already selected tray) or invalid (for example, a Paper Source command selecting a medium beyond the end of the media list).

The behavior of the commands is as follows:

- Paper Source forces the following page to a front side.
- Page Size forces the following page to a front side.
- Duplex Page Side Selection forces the following page to a particular side depending on the command value:
  - “Front” forces the following page to a front side.
  - “Back” forces the following page to a back side.
  - “Next” does not force any particular side.
- Number of Copies causes a multi-set set boundary to occur prior to the affected page, thereby forcing the following page to be the first page of a new set, with the page image appearing on the front side.

Note: A page change that uses a Paper Source or Page Size command in conjunction with a Duplex Page Side Selection command with a value of Back first forces a front side and then a back side. The page is printed on the back side of a sheet with a blank front side. The medium associated with the front side of the sheet is the medium selected by the Paper Source or Page Size command.

A simplex plex exception applied to the *front* of a two-sided page results in the excepted page becoming a simplex page and what would have been the back side becomes the front side of the next sheet. A simplex plex exception applied to the *back* side of a two-sided page puts the back side onto a simplex sheet, creating a blank page.

When a two-sided exception is applied to a simplex page, the page becomes the front side of a two-sided sheet unless that page is preceded by a Duplex Page Side Selection with a value of “Back,” in which case it becomes a back side. If the following page also has a two-sided exception without a side override, it takes the next available side (normally a back side unless the preceding page was placed on a back side in which case the second exception is a front side), unless the page transition is caused by one of the commands that forces a specific side, in which case it becomes the specified side.

Image shifting

For systems that use releases prior to release 7.x, the HorizontalImageShift and VerticalImageShift attributes can be used to alter the placement of the page image on the physical sheet.

Starting with release 7.x, additional image shift attributes are available in support of mixed plex applications.

See the *Guide to Managing Print Jobs* chapter “Document attributes” for a complete explanation of each attribute.

HorizontalImageShiftForBack

This attribute governs the placement of the image, in millimeters, from the edge of the back (or even) page of a duplex job. This attribute can be overridden by the ExceptionPages shift attribute. If the HorizontalImageShiftForBack attribute is not set for a duplex job, the HorizontalImageShift value is used for both sides of the job.

VerticalImageShiftForBack

This attribute governs the placement of the image, in millimeters, from the bottom of the back (or even) page of a duplex job. This attribute can be overridden by the ExceptionPages shift attribute. If the VerticalImageShiftForBack attribute is not set for a duplex job, the VerticalImageShift value is used for both sides of the job.

ExceptionPages Shift Type

This attribute allows a change in image shift for specified pages. The shift type can be HorizontalImageShift or VerticalImageShift for the front (or odd) pages, or HorizontalImageShiftForBack or VerticalImageShiftForBack for back (or even) pages of duplex jobs.

Other attributes that support true mixed plex

There are several other attributes that are available in support of true mixed plex applications.

Uncollated This attribute governs whether or not copies of jobs are collated in the output.

CycleExceptions This attribute can be used for jobs that are made up of many smaller sets of a fixed number of pages. It allows page exceptions, such as plex changes, to be replicated throughout the document without needing the page exceptions to be explicit beyond the first set of pages.

See the *Guide to Managing Print Jobs* chapter “Document attributes” for a complete explanation of each attribute.

Index