The Display PostScript™ System

Adobe ShowPS™
User Guide

Adobe Systems Incorporated

Corporate Headquarters
1585 Charleston Road, PO Box 7900
Mountain View, CA 94039-7900
(415) 961-4400 Main Number
(415) 961-4111 Developer Support

Adobe Systems Eastern Region
24 New England
Executive Park
Burlington, MA 01803
(617) 273-2120

Adobe Systems Europe B.V.
Office Centre
Jozef Israëlskade 48c
1072 SB Amsterdam, Netherlands
31-20-6767-661

Adobe Systems Japan
Swiss Bank House 7F
4-1-8 Toranomon, Minato-ku
Tokyo 105, Japan
03-3437-8950

Version of 15 December 1994

Contents

Introduction

Adobe ShowPS™ is a PostScript™ language document previewer from Adobe Systems Incorporated. You can use it to look at PostScript language documents on the screen before you send them to a printer. Special features allow you to preview documents in a variety of ways:

■ Choose from various standard or custom page sizes.

■ Magnify or reduce the view of the document, or change the orientation of the page as needed.

■ See how a color document would look on a monochrome printer.

■ Look at documents generated by a variety of applications, including Encapsulated PostScript (EPS) files.

To use the previewer, you should have a working knowledge of UNIX® and know how to manipulate windows in the window manager you are running under the X Window System™.

About this manual

Your background and experience determine which chapters of this manual you will find most useful:

■ Chapter 1 gets you started with a tutorial-style introduction to the previewer.

■ Chapter 2 describes all menus, dialog boxes, and windows.

■ Chapter 3 presents a table of all command line options and all X resources, followed by more detailed information about each option and resource.

■ Chapter 4 discusses the *filterps* utility.

■ Appendix A provides troubleshooting information.

■ Appendix B summarizes how the previewer deals with different kinds of documents.

■ Appendix C lists all standard page sizes.

Chapters 1 and 2 are intended for nonprogrammers, while Chapter 3 will be of special interest if you work with UNIX and Motif-based tools every day. Some information is duplicated in different chapters.

System requirements

Adobe ShowPS uses the Display PostScript extension in the X server to display PostScript language documents in an X window. It requires an X11R4 (or later) release of the X Window System, with the Display PostScript system available in the X server.

The Adobe ShowPS product components

The Adobe ShowPS product components are:

■ The Adobe ShowPS executables and UNIX manual page files.

■ The Adobe *filterps* executables and UNIX manual page files.

■ The PostScript language file for this User Guide, which you can print on your PostScript printer.

Registration

To register your copy of Adobe ShowPS, fill out the registration page at the end of this document and fax it to Adobe Systems Incorporated at 1-415-903-0589. You can also register by e-mail by sending the information requested on the card to showps@adobe.com. Your registration will enable Adobe to keep you up to date about new product releases and other important information.

To request sales literature or technical notes by fax, call Adobe at 1-408-986-6587.

For general information about Adobe products and customer service, call 1-800-833-6687 or 1-408-986-6555.

For technical support, contact your workstation vendor.

Chapter 1: *Using the Adobe ShowPS previewer*

This chapter introduces the previewer. By following the procedures you will learn to:

- Start and exit the previewer.
- Open and examine documents.
- Work with several documents.
- Magnify and reduce the page and use other options.

This chapter focuses on frequently used menu commands. Chapter 2 provides a complete discussion of all menu commands.

If the previewer does not respond as described, or if you have any other problems with the previewer, please refer to Appendix A, "Troubleshooting." You may quit at any time by choosing Exit from the File menu.

Starting the previewer

You start the previewer from a command-line prompt using one or more options (see Chapter 3).

To start the previewer and look at a document:

- Type

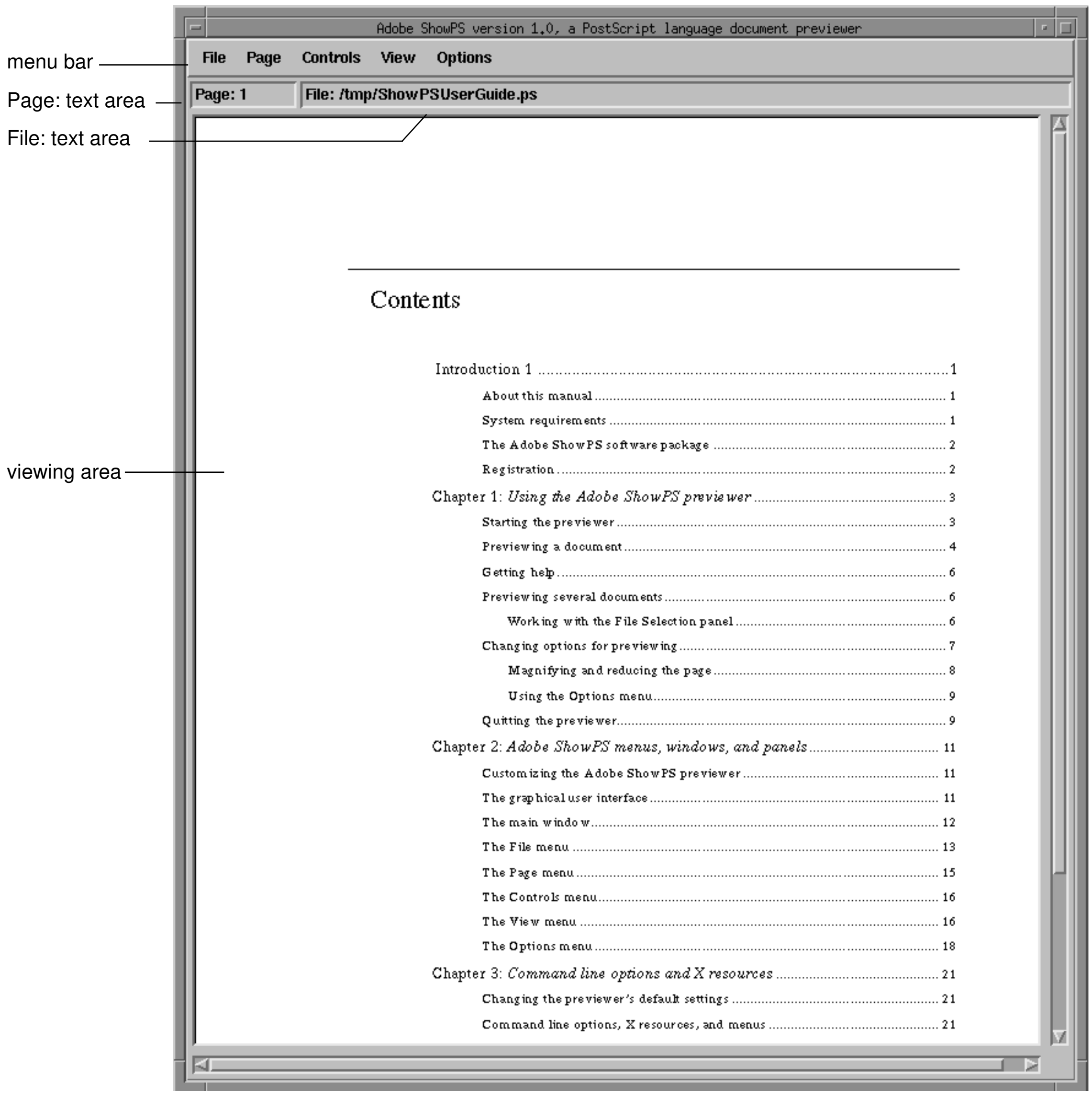
```
% showps filename ...
```

where `%` represents the command-line prompt and *`filename`* is the name of the PostScript language document you want to preview.

The previewer main window appears. Below the menu bar, the Page: text area shows the page number, and the File: text area displays the complete pathname for the file you are currently viewing. By default, the copyright notice is displayed as the first file.

After a short time, the copyright notice disappears and the window displays the first page of the document you requested.

If you want to reshape the window or turn it into an icon, use the commands or mouse gestures of the window manager you are running under X.



Previewing a document

In this section, you learn how to use the Page menu to move forward and backward in a document, and how to go directly to a specified page.

NOTE: This user guide explains how to work with the previewer using menu commands. For most menu commands, a keyboard accelerator is available. It appears on the menu next to the command name. You can use the accelerator when the previewer is the selected application. You don't need to bring up the *menu.*

To page sequentially through a document:

- Choose Next Page from the Page menu to move forward.
- Choose Previous Page from the Page menu to move backward.

Page	
Previous Page	Alt/b
Next Page	Alt/f
Last Page Viewed	Alt/l
Redraw Page	Alt/r
Page Selection...	Alt/p

The number in the Page: text area below the menu bar changes.

Previous Page automatically changes to First Page if the previewer cannot find information about page numbering in the document you are viewing. See Appendix A, “Troubleshooting,” for more information.

You can quickly display a specific page using the Page Selection panel. Note that the previewer uses the number of the page relative to the beginning of the document, not the internal page numbers that may be visible on the page. If, for example, you are looking at a document that starts with page 30, the previewer considers page 45 the 15th page in the document.

To go directly to a specific page in a document:

1. Choose Page Selection from the Page menu.

   The Page Selection panel appears.

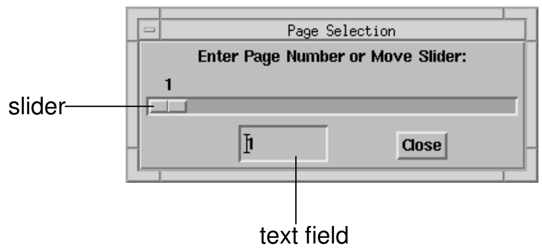
2. In the Page Selection panel, specify a page number in one of these ways:

   - Position the mouse cursor over the slider, then press and hold the left mouse button while moving the slider. The number above the slider changes as you move the mouse. Release the mouse button when the desired number appears.
   - Click in the text field, replace the existing number with the requested number, and press Return.

Sometimes you want to return to a page you just looked at.

To alternate between two pages that are not adjacent:

- Choose the Last Page Viewed command from the Page menu.



This is useful, for example, if you preview an index, then select a page. To return to the index, use Last Page Viewed.

Getting help

To see a summary of command-line options for the Adobe ShowPS previewer:

■ Type

```
showps -help
```

on the command-line prompt.

More complete information is also available:

To look at the UNIX manual page for the previewer:

■ Type

```
man showps
```

on the command-line prompt. The manual page includes information on all command-line options and all X resources.

Previewing several documents

The previewer allows you to load several documents without having to close any of them. Commands in the File menu let you to move back and forth between loaded documents.

You have two options for loading documents:

■ Specify the filenames on the command-line when you start the previewer (see page 3).

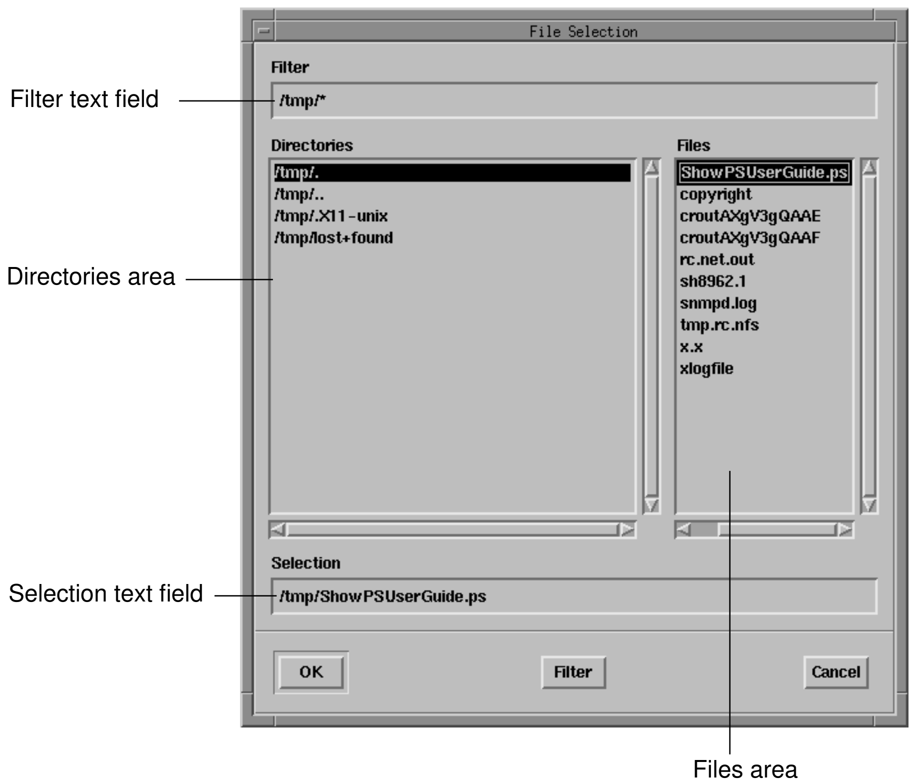
■ Use the File Selection panel, discussed next.

Working with the File Selection panel

To load a file for previewing, choose Open from the File menu. A standard Motif file selection panel appears. You can now specify the file. More advanced users can use filters to search directories and then select the file. This is described in Chapter 2, page 14.

To specify a filename explicitly:

1. Type the pathname into the Selection text field at the bottom of the panel.
2. Click on OK.



Changing options for previewing

You can customize how a document is displayed. For example, you may want to change the orientation of the page for a document using landscape orientation, or see how a color document will look on a monochrome printer. You can usually use an X resource, a command-line option, or a menu command to achieve the same customization:

■ The X resource changes the option default for all future sessions (see Chapter 3).

■ The command-line option overrides the default for one session (see Chapter 3).

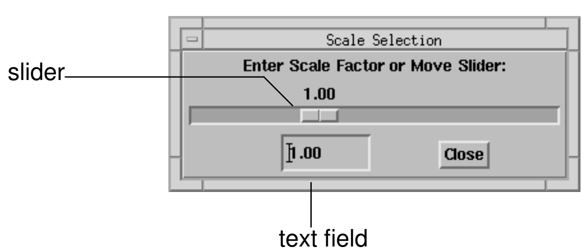
■ The menu command temporarily overrides the default for all loaded documents and all new documents you load, but lets you go back to the default behavior without exiting the previewer.

Magnifying and reducing the page

The previewer lets you set a scale factor to magnify or reduce the pages in a document. This facilitates viewing a document using a small font or a large illustration. For example, if you reduce the scale factor to one half (.5), the previewer displays a half-size page in the upper left corner of the viewing area. If you increase the scale factor to two (2.0), the document appears in the viewing area at twice its original size. Scroll bars allow you to see the rest of a scaled up page.

To bring up the Scale Selection panel:

■ Choose Scale Selection... from the View menu. The Scale Selection panel appears.



Set the scale factor in one of two ways:

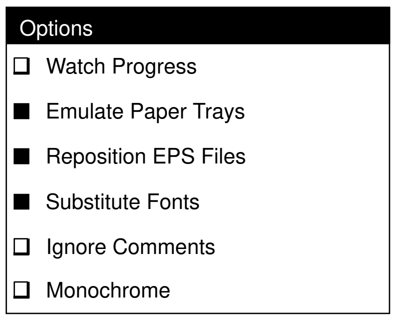
■ Position the mouse cursor over the slider, then press and hold the left mouse button while moving the slider. The number above the slider changes. Release the mouse button at the desired scale factor.

■ Click in the text field, replace the existing number, and press Return.

To change the default range of the slider, type a number into the text field.

NOTE: When you change the scale factor, a message regarding pixmap storage may appear. See page 40 in Appendix A, "Troubleshooting," for more information.

Using the Options menu



The Options menu allows you to change the way the previewer treats documents. For example:

- To prevent the previewer from changing the size of the displayed page when the document asks for a different size paper, turn off Emulate Paper Trays.
- To preview a document as an unstructured document, turn on Ignore Comments. This may allow you to view a document that is structured incorrectly. See Appendix B for more information about structured and unstructured documents.
- To see how a color document would look on a monochrome printer, turn on the Monochrome option.

Each menu command is discussed in detail in Chapter 2.

Quitting the previewer

When you're ready to exit the program, choose Exit from the File menu. All windows that are part of the previewer disappear from screen.

Chapter 2: *Adobe ShowPS menus, windows, and panels*

This chapter first gives an overview of the different interfaces for customizing the previewer. The rest of the chapter provides a complete discussion of the graphical user interface. In contrast to the previous chapter, all menus, windows, and panels are included.

Customizing the Adobe ShowPS previewer

The previewer runs under the X Window System, using the Display PostScript extension in the X server. You can customize the previewer in three ways:

■ Before you start the previewer, you can set X resources. They specify the size and location of panels and the default setting for certain options. They also let you customize keyboard accelerators. Your system administrator sets the resources for your site, but you can override each resource setting in your own resource file. See page 31 in Chapter 3.

■ When you start the previewer, you can customize the current session with command-line options. For example, you can load one or more files or change the default size or orientation of the page. Command-line options override resource settings. See Chapter 3.

■ While the previewer is running, you can override X resource settings and command-line options with menus and dialog boxes.

A table of menu commands, command-line options, and X resources appears at the beginning of Chapter 3 on page 21.

The graphical user interface

The previewer uses a standard Motif user interface:

■ Pull-down menus allow you to load files, page through documents, and set options. If a menu command is followed by an ellipsis (…), selecting it brings up a panel.

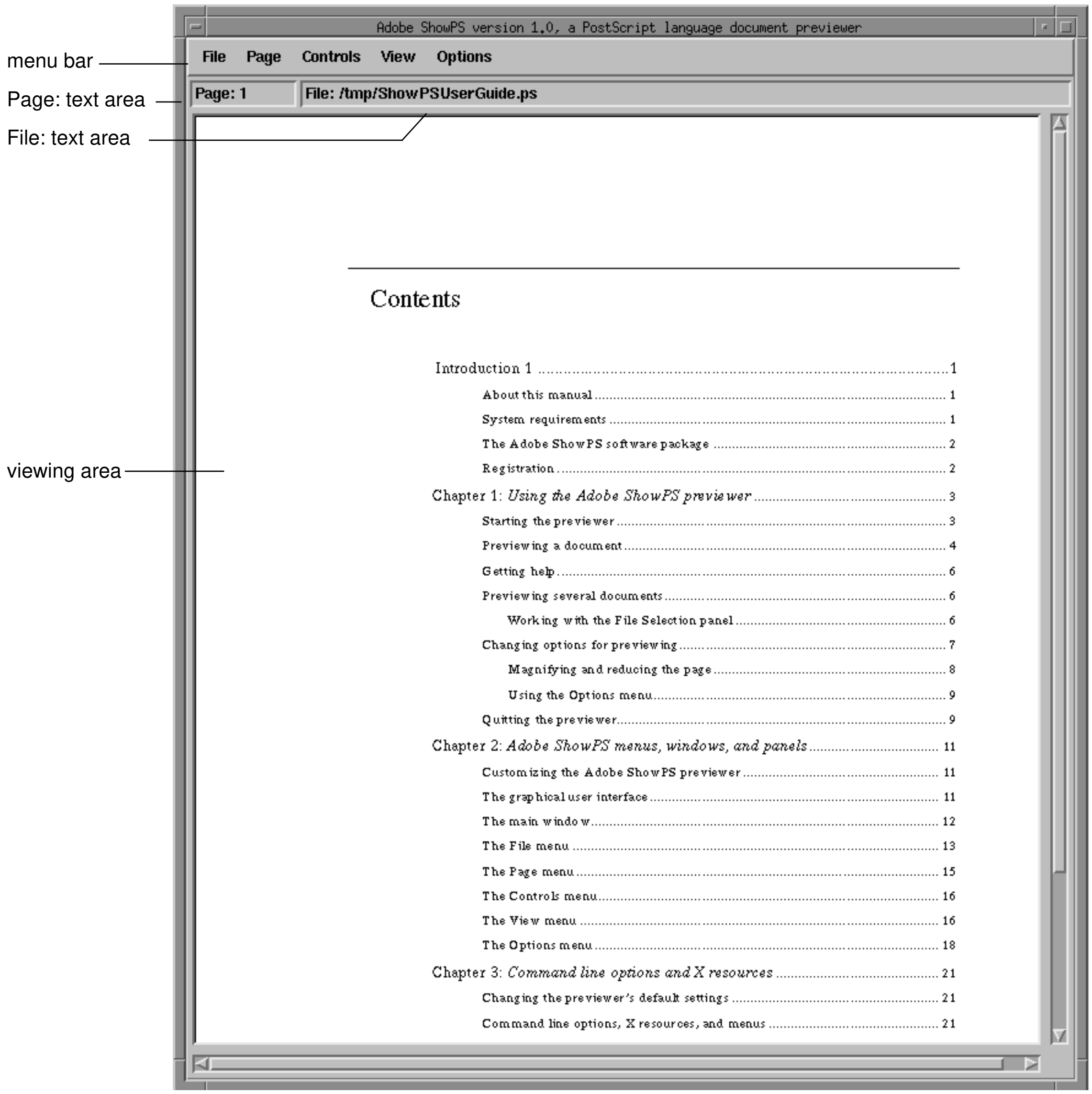
■ Most menu commands have an associated keyboard accelerator. It appears on the menu next to the command name.

■ Each dialog box or panel has a Close or Cancel button.

■ To select files, you use a standard Motif file selection panel.

NOTE: Each window manager has its own methods of moving and resizing windows. Refer to your window manager documentation for information.

The main window



You usually interact with the previewer from the main window. The menu bar allows you to bring up menus. Below the menu bar, the Page: and File: text areas display the page number and the current file. The window manager provides scroll bars and sizing corners to position a page within the window and to change the window size.

The File menu

File	
Previous	Alt/B
Next	Alt/F
Open…	Alt/O
Reopen	Alt/R
Exit	Alt/Q

The File menu lets you load documents, move back and forth between loaded documents, and exit the previewer.

When you load several documents, the previewer creates a trail of available documents you can traverse. As you move from the current document to documents loaded earlier or later, the previewer reopens each document. It opens each document on the first page. To reopen a structured document on the page you viewed last, choose the Reopen menu item. If you return to a document that contained an error, the error recurs unless the file has been corrected.

Previous Displays the document preceding the current document.

Next Displays the document following the current document.

Open… Brings up a standard Motif file selection panel for selecting and loading documents. You can specify the complete filename or search directories for the document you want to view.

To specify a filename explicitly:

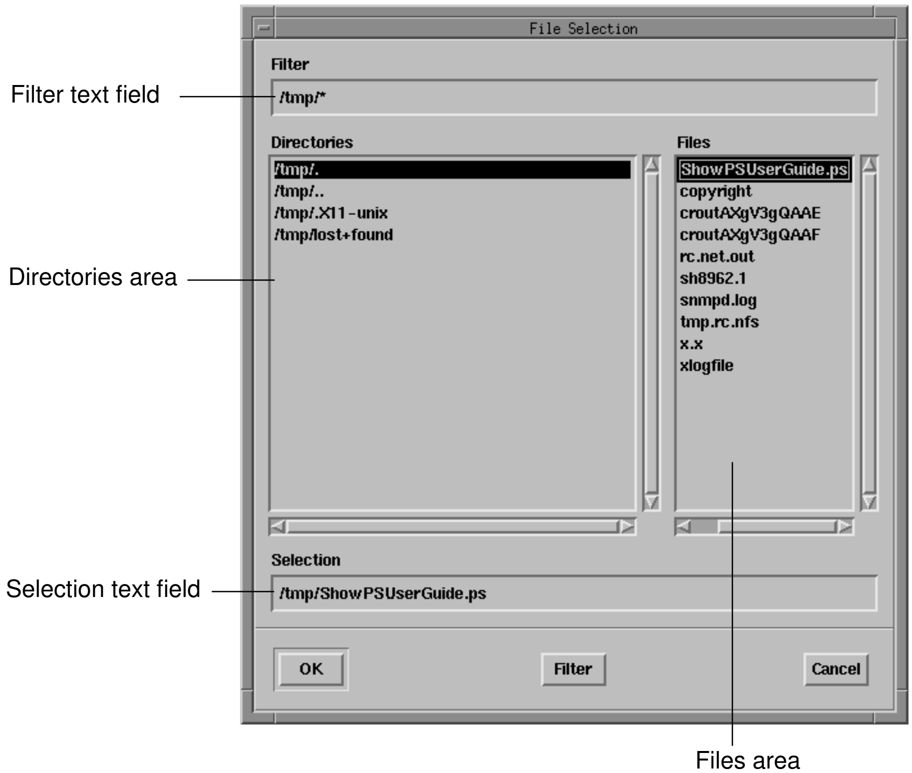
1. Type the pathname into the Selection text field at the bottom of the panel.
2. Click on OK.

Using the File Selection Panel

If you don't know the exact pathname for a document, you can search for it using the Filter text field in the File Selection panel. Wildcard characters in the Filter text field help you narrow down the search. Use an asterisk (*) to represent an arbitrary number of characters, or a question mark (?) for exactly one character. Since PostScript language files have, by convention, an extension of *.ps* or *.PS*, you can filter out files the previewer cannot load. For example:

■ To display all directories and files in the */user/smart* directory, specify */user/smart/**, then click on Filter.

■ To display all files with the extension *.ps* in the same directory, specify */user/smart/*.ps*, then click on Filter.



To search directories and load a file:

1. Type the pathname for the directory you want to search into the Filter text field.
2. Click on Filter.

   All subdirectories of the specified directory appear in the Directories display.
3. To see the directories and files in a subdirectory, click twice on the directory name. This moves the directory name into the Filter text field. Click on Filter again to see the directories and files in that directory.
4. When the file you want to view is visible in the Files display, you can load it in two ways:

   ■ Double-click on the filename in the Files area to select and load the document.

   ■ Select the filename with a single click, then click on OK or press Return to load the document.

   The File Selection panel disappears after you have selected a file. If the file cannot be loaded, the previewer pops up an error message, shows a blank page in the viewing area, and displays the name of the file in the File: text area.

Reopen Reloads the document you are currently viewing. This is useful if you have changed a document and want to see the result. If the document is structured, it will reopen on the current page; otherwise, it will reopen on the first page.

Exit Exits the previewer.

The Page menu

Page	
Previous Page	Alt/b
Next Page	Alt/f
Last Page Viewed	Alt/l
Redraw Page	Alt/r
Page Selection…	Alt/p

The Page menu lets you browse through a document or select a specific page using the Page Selection panel (page 15).

When you load an *unstructured* document (a document that does not contain Adobe Document Structuring Convention comments), the previewer changes the menu. See Appendix B for more information. Menu commands are grayed when they are unavailable. For example, when you preview the first page of a document, Previous Page is grayed.

Previous Page Displays the previous page of a document. If you are viewing an unstructured document, the command changes to First Page.

Other accelerators: p (previous), b (back).

Next Page Displays the next page of a document. This command is not available when you are on the last page of a document.

Other accelerators: n (next), f (forward), Return, spacebar

Last Page Viewed Displays the page you viewed before viewing the current page. Use this command to return to a nonadjacent page. For example, you may look at an index, then select a page. Last Page Viewed returns you to the index.

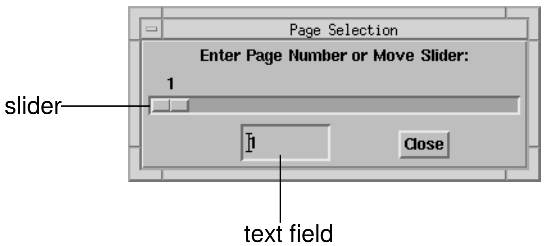
Redraw Page Redraws the currently selected page. This command is available only when you are viewing a structured document.

Page Selection… Pops up the Page Selection panel. There are two ways to select a page:

- Position the mouse cursor over the slider and press and hold the left mouse button while moving the slider. The number above the slider changes as you move the mouse. Release the mouse button to make your selection.
- Click in the text field, replace the number, and press Return.

The Page Selection panel remains available until you close it. While the panel is on screen, the menu command changes to Close Page Selection Panel.

This command is grayed if you are viewing an unstructured document.



The Controls menu

Controls	
Interrupt	Alt/I
Continue	Alt/O

The Controls menu lets you interrupt the previewer as it displays a page. After you have interrupted the previewer, you can use Continue to complete displaying the same page. You can also request a different page or open a different document. This is useful if you are loading a large graphics file and decide to abandon the process because it is too time consuming.

Interrupt Interrupts the previewer as it displays a page.

Continue Continues a previously interrupted display of a page.

The View menu

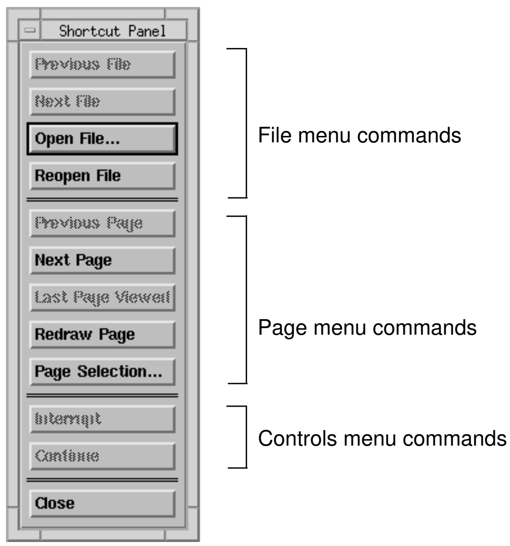
The View menu lets you change the orientation and scale factor for a page and bring up a shortcut panel. Each command in the menu is followed by an ellipsis (…), which indicates that it brings up a panel.

View	
Shortcut Panel…	Alt/c
Scale Selection…	Alt/s
Page Orientation…	Alt/o

When you change the way a document is displayed, the change remains in effect for that session until you explicitly change the same aspect of the display again. For example, if you change the scale factor, the new factor applies to the current document, any new document you load, and any previously loaded document you view again until you change the scale factor again.

When you select a command in the View menu, a panel appears. You can resize or move the panel, and close it when you no longer need it. While any of the panels is on screen, the associated View menu command changes to let you close the panel; for example, Shortcut Panel becomes Close Shortcut Panel.

Shortcut Panel… Provides access to most menu commands. If you're not using the keyboard accelerators, this panel speeds up your interaction with the previewer because you don't have to bring up a menu each time you want to select a command.

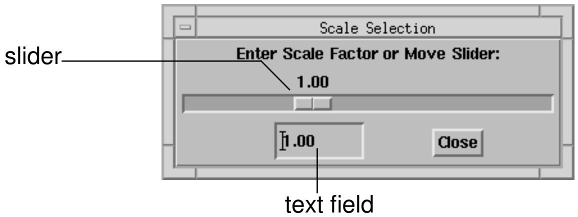


Scale Selection… Allows you to magnify or reduce the displayed page. The size of the window does not change. Choose the scale factor in one of two ways:

■ Position the mouse cursor over the slider and press and hold the left mouse button while moving the slider. The number above the slider changes as you move the mouse. Release the mouse button to make a selection.

■ Click in the text field, replace the existing number, and press Return.

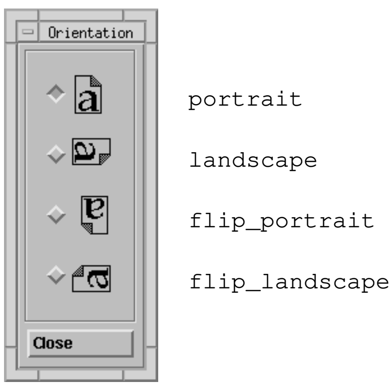
The initial range of the slider is always between half and double the initial scale factor. For the default scale factor (1), the initial range is 0.5 to 2.0. If, for example, you use the `scale` X resource or the `-sc` command-line option to set the initial scale factor to 2.0, the initial range of the slider changes to 1.0 to 4.0.



You can increase the slider range by typing a larger or smaller number into the text field. Numbers between 0.1 and 10.0 will be accepted. The slider range will not return to the initial range during the current previewer session.

NOTE: When you change the scale factor, a message regarding pixmap storage may appear. See page 40 in Appendix A, "Troubleshooting," for more information.

Page Orientation… Displays the Page Orientation panel, which lets you change the orientation of the page within the window. The four choices are illustrated by icons; the text to the right of the icon represents the associated value for the `-or` command-line option:

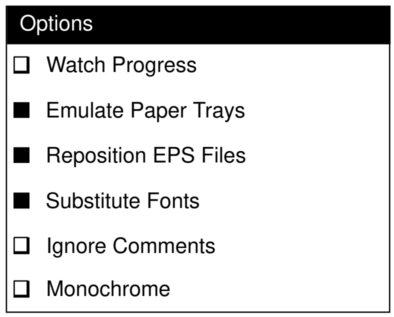


The Options menu

Each command on the Options menu is associated with a button for turning the option on or off. The default for each option is determined by the associated X resource or command-line option.

If you change the Monochrome or Watch Progress option, the new setting affects the current document immediately. If you change one of the other options, you have to reopen the document using Reopen in the File menu.

The figure below shows the standard default for each command.



Watch Progress The previewer normally writes each page first to an offscreen buffer called a pixmap. When the whole page is in the buffer, it is displayed. If you turn on Watch Progress, the previewer writes parts of the page to the window as the pixmap is being prepared. While this option lets you observe the progress of a page that requires lengthy preparation, the order in which parts are displayed is not meaningful.

Emulate Paper Trays When this option is turned on, the previewer changes the size of the display area each time the document contains a request for a different paper tray.

Reposition EPS Files Encapsulated PostScript (EPS) files can contain images at an arbitrary location on the page. By default, the previewer aligns the images so the top left corner of the image is in the top left corner of the page to eliminate unnecessary scrolling. To see the actual location of the image, turn the option off.

Substitute Fonts The previewer normally substitutes Courier font for fonts unavailable in your environment. If you turn this option off, the previewer does not substitute Courier font for unavailable fonts and cannot display documents that require unavailable fonts.

Ignore Comments The previewer normally uses all document structuring convention comments it finds in a document. If a document uses invalid document structuring convention (DSC) comments, the previewer cannot display it. When you turn this option on, the previewer ignores all DSC comments and lets you preview an incorrectly structured document.

Monochrome The previewer normally displays documents in color wherever color is specified. This option allows you to display color documents as they will look on a monochrome printer.

Chapter 3: *Command-line options and X resources*

This chapter starts with a table that lists each command-line option and the corresponding X resource, including its type and default. The sections that follow discuss each command-line option and each X resource in more detail.

Changing default settings

You can change the default behavior of ShowPS by setting an X resource, providing a command-line option, or using a menu command.

- Use the X resource to permanently set your preferred default.
- Use the command-line option to override the resource setting for one session.
- Use the menu command to temporarily override the resource setting or command-line option (see Chapter 2).

X resources and command-line options

X resource	Command-line option	Type, values, and defaults	Description
	`-help, -usage`		Lists command-line options and X resources.
	`-version`		Prints out the ShowPS version.
`colormapRetainColors`	`-retain`	Type: `boolean` Default: `False`	If this resource is set to True and ShowPS has allocated colors in the default colormap, those colors will persist (remain allocated in the colormap) after ShowPS exits. If the visual is not the default visual, or if ShowPS creates a separate colormap, then the allocated colors will *not* be retained, regardless of the value of this resource or option.
`colormapAllowSeparateMap`	`-sepcmap`	Type: `boolean` Default: `False`	Allows ShowPS to create a separate colormap if the Display PostScript colorcube and grayramp do not fit in the default colormap.
`colormapReds`	`-reds` *number*	Type: `integer` Values: `0, 2–256` Default: `0`	Sets the preferred number of red values to use in the Display PostScript colorcube.
`colormapGreens`	`-greens` *number*	Type: `integer` Values: `0, 2–256` Default: `0`	Sets the preferred number of green values to use in the Display PostScript colorcube.
`colormapBlues`	`-blues` *number*	Type: `integer` Values: `0, 2–256` Default: `0`	Sets the preferred number of blue values to use in the Display PostScript colorcube.

X resource	Command-line option	Type, values, and defaults	Description
`colormapGrays`	`-grays` *number*	Type: `integer` Values: `0`, `2–256` Default: `0`	Sets the preferred number of gray values to use in the Display PostScript grayramp.
`displayShortcutPanel-DialogAtStartup`	`-dis1`	Type: `boolean` Default: `False`	Displays Shortcut Panel at startup. The Shortcut Panel contains commonly used File, Page, and Controls menu commands.
`displayScaleSelection-DialogAtStartup`	`-dis2`	Type: `boolean` Default: `False`	Displays Scale Selection panel at startup. The Scale Selection panel lets you enlarge or reduce the displayed image.
`displayPageSelection-DialogAtStartup`	`-dis3`	Type: `boolean` Default: `False`	Displays Page Selection panel at startup. The Page Selection panel lets you select a page for viewing.
`displayPageOrientation-DialogAtStartup`	`-dis4`	Type: `boolean` Default: `False`	Displays the Page Orientation panel at startup. The Page Orientation panel lets you change the orientation of the page within the window.
`emulatePaperTrays`	`-ept`	Type: `boolean` Default: `True`	Changes the size of the viewing area when the document asks for a different paper size.
`ignoreComments`	`-igc`	Type: `boolean` Default: `False`	Ignores document structuring convention comments, previewing documents as unstructured documents.
`maxPixmapBytes`	`-maxp` *size*	Type: `integer` Default: 4 Mbytes	Sets the maximum number of bytes of memory to allow for pixmap storage for the drawing window.
`monochrome`	`-mono`	Type: `boolean` Default: `False`	Shows a document as it will look when printed on a monochrome printer.
`pageOrientation`	`-or` *orientation*	Type: `string` Default:`portrait`	Rotates the page in increments of 90 degrees.
`pageSize`	`-ps` *pagesize*	Type: `string` Default: `letter`	Specifies a standard or custom page size. For custom page sizes, use *width*:*height*.
`preScanSize`	`-pscan` *size*	Type: `integer` Default: `8200`	Determines the size of the beginning of a file that ShowPS searches for the PostScript identifier string “%!”. ShowPS starts rendering the file at the offset where the identifier is found.
`prologFile`	`-pro` *file*	Type: `string` Default: *none*	Sets the pathname of an external PostScript prolog file to be sent to the context after the internal prolog file.

X resource	Command-line option	Type, values, and defaults	Description
`quietFallback`	`-qfb`	Type: `boolean` Default: `True`	Falls back to unstructured mode without giving alert message.
`readFromStandardInput`	-	Type: `boolean` Default: `False`	Reads from *stdin* instead of a file.
`repositionEPSF`	`-repsf`	Type: `boolean` Default: `True`	Aligns the upper left corner of an EPSF image with the upper left corner of the viewing area.
`scale`	`-sc` *`scale`*	Type: `float` Default: `1.0`	Reduces or magnifies the view of the document.
`substituteFonts`	`-sf`	Type: `boolean` Default: `True`	Replaces unavailable fonts with Courier.
`skipCopyright`	`-skipc`	Type: `boolean` Default: `False`	Skips the copyright notice at startup.
`tempDir`	`-tdir` *`directory`*	Type: `string` Default: `/tmp`	Writes files to this directory (and later removes them) when ShowPS reads from *stdin*.
`verbose`	`-v`	Type: `boolean` Default: `False`	Prints a short message to *stderr* each time an error dialog is displayed.
`visualClass`	`-visual` *`visual`*	Type: `string` Values: `Best`, `Default`, `StaticGray`, `GrayScale`, `StaticColor`, `PseudoColor`, `TrueColor`, `DirectColor` Default: `Default`	Sets the visual class for ShowPS to use. See Appendix D on page 49 for an explanation of how ShowPS handles colors.
`visualDepth`	`-depth` *`depth`*	Type: `integer` Values: `1`, `2`, `4`, `8`, `12`, `24` Default: 24	Sets the preferred visual depth for ShowPS to use. See Appendix D on page 49 for an explanation of how ShowPS handles colors.
`watchProgress`	`-wp`	Type: `boolean` Default: `False`	Displays partially complete pages while loading a file.
	`files`		Loads all specified files in sequence.

Command-line options

Command-line options let you customize ShowPS when you start it. You can type the options in upper or lower case. Options may appear in any order, but all options must precede the first filename. ShowPS lets you use the standard X toolkit options, which are listed in the UNIX manual page *X(1)*. In addition, ShowPS supports the following options:

```
showps [-help] [-usage] [-version] [-retain] [-sepcmap]
[-reds number] [-greens number] [-blues number]
[-grays number] [-dis1] [-dis2] [-dis3] [-dis4] [-ept] [-igc]
[-maxp size] [-mono] [-or orientation] [-ps pagesize]
[-pscan size] [-pro file] [-qfb] [-] [-repsf] [-sc scale] [-sf]
[-skipc] [-tdir directory] [-v] [-visual visual]
[-depth depth] [-wp] [files]
```

`-help`
`-usage` Lists command syntax, command-line option summary, and X resource for each option. The `-help` and `-usage` command options are equivalent.

`-version` Displays the version number of the ShowPS product.

`-retain` (retain colors) If ShowPS allocates colors in the default colormap, those colors, by default, do *not* remain allocated in the colormap after ShowPS exits. Instead, the entries are freed and made available for other applications that use the default colormap. However, if the `-retain` option is specified and ShowPS has allocated colors in the default colormap, those colors *do* remain allocated in the colormap after ShowPS exits.

If the visual is not the default visual, or if ShowPS creates a separate colormap, then the allocated colors are *not* retained, whether or not the `-retain` option is specified and regardless of the value of `colormapRetainColors`, the associated X resource. `colormapRetainColors` is `False` by default (see table on page 21).

`-sepcmap` (separate colormap) The associated X resource for this option is `colormapAllowSeparateMap` (see table on page 21).

If the visual is *not* the default visual, ShowPS allocates the Display PostScript colorcube and grayramp in a separate colormap whether or not the `-sepcmap` option is specified and regardless of the value of `colormapAllowSeparateMap`.

If the visual *is* the default visual, ShowPS first tries to allocate the Display PostScript colorcube and grayramp in the default colormap. If the colorcube and grayramp fit in the default colormap, then no further action is taken. However, if there is not enough room, then ShowPS either shrinks the colorcube and grayramp until they fit in the default colormap or allocates a separate colormap, according to the following rules:

ShowPS shrinks the colorcube and grayramp if either of the following conditions is true:

- The selected visual is read-only (`StaticGray`, `StaticColor`, or `TrueColor`).
- The `colormapAllowSeparateMap` resource is set to False and the `-sepcmap` option is *not* specified.

If neither of these conditions is true, ShowPS allocates the Display PostScript colorcube and grayramp in a separate colormap.

Note that allowing ShowPS to create a separate colormap can cause other applications to be shown in false colors when the window manager's focus is on ShowPS. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`-reds` *`number`* Sets the preferred number of red colors in the Display PostScript colorcube. Values of *`number`* can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors. The associated X resource for this option is `colormapReds` (see table on page 21).

`-greens` *`number`* Sets the preferred number of green colors in the Display PostScript colorcube. Values of *`number`* can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors. The associated X resource for this option is `colormapGreens` (see table on page 21).

`-blues` *`number`* Sets the preferred number of blue colors in the Display PostScript colorcube. Values of *`number`* can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors. The associated X resource for this option is `colormapBlues` (see table on page 21).

`-grays` *`number`* Sets the preferred number of grays in the Display PostScript grayramp. Values of *`number`* can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors. The associated X resource for this option is `colormapGrays` (see table on page 22).

`-dis1` (display option 1) Displays the Shortcut Panel at startup for immediate access. The Shortcut Panel contains commonly used File, Page, and Controls menu commands. The associated X resource, `displayShortcutPanelDialogAtStartup`, is `False` by default.

`-dis2` (display option 2) Displays the Scale Selection panel at startup. The Scale Selection panel lets you enlarge or reduce the displayed image. The associated X resource, `displayScaleSelectionDialogAtStartup`, is `False` by default.

`-dis3` (display option 3) Displays the Page Selection panel at startup. The Page Selection panel lets you select a page for viewing. The associated X resource, `displayPageSelectionDialogAtStartup`, is `False` by default.

`-dis4` (display option 4) Displays the Page Orientation panel at startup. The Page Orientation panel lets you change the orientation of the page within the window. The associated X resource, `displayPageOrientationDialogAtStartup`, is `False` by default.

`-ept` (emulate paper trays) Changes the page size when the PostScript language document contains a request for a different paper size. The associated X resource, `emulatePaperTrays`, is `True` by default. You need this option only if the resource has been changed to `False` and you want to temporarily change the behavior.

`-igc` (ignore comments) Ignores all DSC comments in a PostScript language file, that is, treats the document as an unstructured document. This can be helpful for viewing an incorrectly structured document. See Chapter 4 for another way of dealing with incorrectly structured documents.

The associated X resource for this option is `ignoreComments` (see table on page 22.)

`-maxp` *`size`* When ShowPS displays a page, it first writes it to an offscreen buffer called a pixmap. This option sets the maximum number of bytes of memory ShowPS is allowed to allocate for pixmap storage. The default value is 4 Mbytes. The associated X resource for this option is `maxPixmapBytes` (see table on page 22.)

The *size* parameter may be followed by `k` for Kbytes and `m` for Mbytes. If *size* is greater than 100 Mbytes, an error message results. If *size* is zero, ShowPS does not allocate a pixmap but draws directly to the window instead.

If you increase the page size or the scale factor, the memory required for the pixmap may exceed the maximum ShowPS is allowed to allocate. In that case, a message informs you that the page will now be drawn directly to the window. Click OK to continue. For more information, see page 40 in Appendix A, “Troubleshooting.”

Example: To start ShowPS and set a maximum pixmap size of 5 Mbytes:

```
% showps -maxp 5m
```

`-mono` Allows you to preview a document as it would look if you printed it on a monochrome printer, even if it is a color document. Using this option may improve performance of ShowPS, since monochrome documents require a much smaller pixmap. You can choose between optimum performance and optimum display quality:

- Consider using `-mono` to enhance performance if you are previewing monochrome documents on a color display.
- Don't use `-mono` if memory or speed are not an issue; this gets you the highest quality display on all monitors.

The associated X resource for this option is `monochrome` (see table on page 22).

`-or` *`orientation`* Sets the page orientation to the specified value. Valid choices are:

- `portrait`—The "normal" upright orientation
- `landscape`—The page rotated 90 degrees clockwise
- `flip_portrait`—The page rotated 180 degrees clockwise
- `flip_landscape`—The page rotated 270 degrees clockwise

The Page Orientation panel, page 17, illustrates the four options. The associated X resource for this option is `pageOrientation` (see table on page 22).

Example: To start ShowPS and rotate all pages clockwise by 90 degrees:

```
% showps -or landscape
```

`-ps` *`pagesize`* Specifies the size of the drawing area. *`pagesize`* can be the name of a standard page size (ledger, b, c0, and so on) or two numbers, separated by a colon, that define the width and height of a custom page. The default page size is 8.5 × 11 inches.

Possible values for the pagesize argument include:

- a, b, c, d, e, f, or e1 (American ANSI)
- a0, a1, … a10 (metric ISO)
- b0, b1, … b10 (metric ISO)
- c0, c1, … c7 (metric ISO)
- width:height (custom size)

Width and height are by default specified in inches and may be integers or floating-point numbers. To specify units other than inches, follow the number with:

- cm to specify centimeters
- mm to specify millimeters
- px to specify pixels
- pt to specify points

Other values for the *pagesize* argument are supported for convenience, such as legal (8.5 × 14), ledger (11 × 17), and executive (7.25 × 10.50).

Case is significant when specifying arguments to the `-ps` option. In many instances, upper-case arguments designate 90 degree rotated versions of the lower case arguments. For example, “letter” designates a page size of 8.5 × 11, while “Letter” designates a page size of 11 × 8.5. See Appendix C for a complete list of possible values for the *pagesize* argument.

The associated X resource for this option is `pageSize` (see table on page 22). Appendix C lists the dimensions of all standard page sizes for reference.

If you choose a page size too small for the page you are previewing, the effect is the same as using a paper size that is too small for a document in a printer. ShowPS displays as much of the document as fits on the page but does not provide scroll bars. To view the whole page, choose a large enough page size, then scroll or scale the document as needed if it is larger than the window.

Examples:
To start ShowPS for viewing the *big.ps* document, which is 30 cm high and 20 cm wide:

```
%showps -ps 20cm:30cm big.ps
```

To start ShowPS and set up the viewing area for legal size documents:

```
%showps -ps legal
```

`-pscan size` (pre-scan size) Certain applications prepend sequences of control characters and other text strings to the PostScript files they generate. When ShowPS opens such a file, it searches through the first *size* characters for the PostScript identifier string ``%!". If found, ShowPS starts rendering the PostScript file starting at the file offset where the identifier was found. If not found, ShowPS assumes that the identifier starts at the character immediately following the first *size* characters of the file. If the identifier string is missing, ShowPS displays a dialog box that asks whether to display the file anyway. The *size* parameter may be followed by `k` for Kbytes and `m` for Mbytes. If *size* is greater than 100 Mbytes, an error message results. If *size* is zero, ShowPS skips the pre-scan and assumes that the PostScript identifier string starts in the first position in the first line of the file. The default value is 8200, which allows for the common case where the header size is 8192 bytes.

`-pro file` Sets the pathname to an external PostScript prolog file. ShowPS sends this file to the Display PostScript context after the internal prolog and before the PostScript file.

This option can be used to implement stubs for site-specific **statusdict** operators, which may be used in PostScript files at certain sites.

The associated X resource for this option is `prologFile` (see table on page 22).

`-qfb` (quiet fall back) When set, ShowPS falls back to unstructured mode if it cannot find %%Page comments in the PostScript file being viewed. When not set, a message window appears to alert the user that the file is not structured.

`-repsf` (reposition EPS files) In an Encapsulated PostScript (EPS) file, an image can be anywhere on the page. By default, ShowPS aligns the upper left corner of the image with the upper left corner of the page. The associated X resource (`repositionEPS`) that determines this behavior is `True` by default (see table on page 23). If the resource has been set to `False`, you can use the `-repsf` option to temporarily return to the default behavior.

`-sc scale` Sets a scale factor which applies to the page size and to the drawing on the page. The scale factor may be an integer or a floating-point number. The default value for the scale factor is 1 (one). You can use numbers between 0.1 and 10.0. The associated X resource for this option is `scale` (see table on page 23).

If you use this option, the slider in the Scale Selection panel will have a different initial range because the initial range lets you view pages at half to twice the size at startup. For the default scale factor of 1, the initial slider range is 0.5 and 2.0. If, for example, you use a scale factor of 2, the slider range is between 1.0 and 4.0

Example: To start ShowPS, load the *big.ps* document, use a legal size page and reduce the view to 60%:

```
%showps -ps legal -sc .6 big.ps
```

`-sf` (substitute fonts) Causes ShowPS to substitute Courier font if the file calls for a font that is not available.

You need this option only if the value of the `substituteFonts` X resource has been set to `False` (see table on page 23). In that case, ShowPS displays an error message if you try to load a file that requires unavailable fonts and the X server does not provide font substitution. The message explains why ShowPS cannot load the file.

`-skipc` (skip copyright) Makes ShowPS skip the copyright notice which by default is displayed when you start the program. The associated X resource for this option is `skipCopyright` (see table on page 23).

`-tdir` *`directory`* When the program reads from *stdin*, it temporarily stores the input from *stdin* in the directory specified by the `tempDir` X resource (*tmp* by default). The associated X resource for this option is `tempDir` (see table on page 23). Use this option to specify temporarily a different directory for storing the input.

Example: To start ShowPS and have it store files in */user/docs/temp* when it reads from *stdin:*

```
%showps -tdir /user/docs/temp
```

`-v` (verbose) When you use this option, the program prints a short message to *stderr* each time an error dialog box is displayed. This allows you to save error messages to a file for later inspection and debugging. The associated X resource for this option is `verbose` (see table on page 23).

`-visual` *`visual`* (visual class to use) Directs ShowPS to use the specified visual class.

Values are: `Best`, `Default`, `StaticGray`, `GrayScale`, `StaticColor`, `PseudoColor`, `TrueColor`, `DirectColor`.

The default visual class is `Default`. The associated X resource for this option is `visualClass` (see table on page 23). See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`-depth` *`depth`* Sets the preferred depth of the visual. Values for *`number`* can be integers 1, 2, 4, 8, 12, and 24. Default value is 24. The associated X resource for this option is `visualDepth` (see table on page 23). See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`-wp` (watch progress) When ShowPS prepares a page for display, it normally writes it first to an offscreen buffer called a pixmap. When the whole page is in the buffer, it is displayed. If you use this option, ShowPS writes parts of the page to the window as the display is being prepared. Although this option lets you observe the progress of a display, the order in which parts are displayed is not meaningful. The associated X resource for this option is `watchProgress` (see table on page 23).

– *files* Allows you to load one or more files into ShowPS at startup time. If you supply "`–`" instead of a filename, the program reads from *stdin*. In that case, the program temporarily stores the input from *stdin* in the directory specified by the *tempDir* X resource and later removes it.

The "`–`" option makes it possible to view plain text files. Run the file through a program that generates PostScript language output and pipe the output to the *stdin* of ShowPS.

Adobe ShowPS X resources

This section lists the application-specific X resources for ShowPS. The default and type for each resource is included in the table on page 21.

> *NOTE: If you have no experience setting X resources, use the command-line options or menu commands to change ShowPS defaults.*

The application defaults file for ShowPS is *Showps*. Only the system administrator should change this file.

Many of the widgets used by ShowPS get their resources from *Showps*. Because of that, *Showps* must be in one of the directories specified by the `XFILESEARCHPATH` environment variable. To override any of the resources, set them in your *.Xdefaults* file.

The class name for the Adobe ShowPS application is `Showps`. Thus, to permanently set the `monochrome` resource for ShowPS, add the following line to your *.Xdefaults* file:

```
Showps.monochrome:True
```

ShowPS understands standard Motif resource names and classes. In addition, the following resources allow application-specific customization:

`colormapRetainColors`

If ShowPS allocates colors in the default colormap, those colors, by default, do *not* remain allocated in the colormap after ShowPS exits. Instead, the colormap entries are freed and made available for other applications that use the default colormap. However, if the `colormapRetainColors` resource is set to True and ShowPS has allocated colors in the default colormap, those colors remain allocated in the colormap after ShowPS exits. If the visual is not the default visual, or if ShowPS creates a separate colormap, then the allocated colors are *not* retained, regardless of the value of the `colormapRetainColors` resource or the specification of the associated command-line option, `–retain` (see table on page 21). See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`colormapAllowSeparateMap`

The associated command line option for this resource is `-sepcmap` (see table on page 21).

If the visual is *not* the default visual, ShowPS allocates the Display PostScript colorcube and grayramp in a separate colormap whether or not the `-sepcmap` option is specified and regardless of the value of `colormapAllowSeparateMap`.

If the visual *is* the default visual, ShowPS first tries to allocate the Display PostScript colorcube and grayramp in the default colormap. If the colorcube and grayramp fit in the default colormap, then no further action is taken. However, if there is not enough room, then ShowPS either shrinks the colorcube and grayramp until they fit in the default colormap or allocates a separate colormap, according to the following rules:

ShowPS shrinks the colorcube and grayramp if either of the following conditions is true:

- The selected visual is read-only (`StaticGray`, `StaticColor`, or `TrueColor`).
- The `colormapAllowSeparateMap` resource is set to False and the `-sepcmap` option is *not* specified.

If neither of these conditions is true, ShowPS allocates the Display PostScript colorcube and grayramp in a separate colormap.

Note that allowing ShowPS to create a separate colormap can cause other applications to be shown in false colors when the window manager's focus is on ShowPS. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`colormapReds` Sets the preferred number of red colors in the Display PostScript colorcube. Values for the argument can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`colormapGreens` Sets the preferred number of green colors in the Display PostScript colorcube. Values for the argument can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`colormapBlues` Sets the preferred number of blue colors in the Display PostScript colorcube. Values for the argument can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`colormapGrays` Sets the preferred number of gray values in the Display PostScript grayramp. Values for the argument can be integers 0, 2–256. The default value is 0. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`displayShortcutPanelDialogAtStartup`

Instructs ShowPS to open the Shortcut Panel at startup. The Shortcut Panel contains commonly used File, Page, and Controls menu commands.

`displayScaleSelectionDialogAtStartup`
: Instructs ShowPS to open the Scale Selection panel at startup. The Scale Selection panel lets you enlarge or reduce the displayed image.

`displayPageSelectionDialogAtStartup`
: Instructs ShowPS to open the Page Selection panel at startup. The Page Selection panel lets you select a page for viewing.

`displayPageOrientationDialogAtStartup`
: Instructs ShowPS to open the Page Orientation panel at startup. The Page Orientation panel lets you change the orientation of the page within the window.

`emulatePaperTrays` Instructs ShowPS to change the size of the displayed page when the document contains requests for paper tray changes. When set to `False`, the program does not honor paper tray requests.

`ignoreComments` Instructs ShowPS to ignore any DSC comments it finds in a document and to treat the document as unstructured.

`maxPixmapBytes` Determines the maximum number of bytes used for pixmap storage.

Unlike the `–maxp` option (see page 26), the value of this resource must be an integer. Scale factors (`m` and `k`) cannot be used.

`monochrome` Instructs ShowPS to display documents in monochrome even on a color display. This simulates the effect of printing on a monochrome printer.

`pageOrientation` Sets the initial orientation of a previewed page. Possible settings are: `portrait`, `landscape`, `flip_portrait`, `flip_landscape`. Default is `portrait`.

`pageSize` Determines the default page size. See Appendix C for a list of standard page sizes. Default is `letter` size.

`preScanSize` Searches file for PostScript identifier string "%!" and starts rendering the PostScript file at the file offset where the identifier is found. If not found, ShowPS assumes that the identifier starts at the character immediately following the first `preScanSize` characters of the file. If the identifier string is missing, ShowPS displays a dialog box that asks whether to display the file anyway. Unlike the `–pscan` option, this resource must be an integer. Scale factors (`m` and `k`) cannot be used. If `preScanSize` is greater than 100 Mbytes, an error message results. If `preScanSize` is zero, ShowPS skips the pre-scan and assumes that the PostScript identifier string starts in the first position in the first line of the file. The default value (8200) allow for the common case in which the header size is 8192 bytes.

`prologFile` Sets the pathname to an external PostScript prolog file. ShowPS sends this file to the Display PostScript context after the internal prolog and before the PostScript file. This resource can be used to implement stubs for site-specific **statusdict** operators, which may be used in PostScript files at certain sites.

`quietFallback` When set to `True`, ShowPS falls back to unstructured mode if it cannot find %%Page comments in the PostScript file being viewed. When not set (or set to `False`), a message window appears to alert the user that the file is not structured.

`readFromStandardInput`

Determines whether ShowPS reads from *stdin* by default, or only when the "–" option is used.

If the "–" option is used to read from *stdin*, or if the `readFromStandardInput` resource is True, the program may have to allocate a temporary file while processing *stdin*. By default, the file is placed in */tmp* and deleted automatically when it is no longer needed. Use the `tempDir` resource to change the default directory.

`repositionEPSF` Determines whether the program places the upper left corner of an EPS image in the upper left corner of the page.

`scale` Sets the initial scale factor ShowPS uses to display documents.

`substituteFonts` Instructs ShowPS to substitute Courier for any missing fonts.

`skipCopyright` Instructs ShowPS to bypass the display of the copyright notice upon startup.

`verbose` Prints a short message to *stderr* each time an error dialog is displayed.

`visualClass` Directs ShowPS to use the specified visual class. Valid values are `Best`, `Default`, `StaticGray`, `GrayScale`, `StaticColor`, `PseudoColor`, `TrueColor`, or `DirectColor`. The default visual class is `Default`. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`visualDepth` Sets the preferred depth of the visual. Valid values for the visual are integers 1, 2, 4, 8, 12, and 24. Default value is 24. See Appendix D on page 49 for an explanation of how ShowPS handles colors.

`watchProgress` Instructs ShowPS to update the display while rendering is in progress instead of waiting for a page to complete before displaying any of it.

Display PostScript NX X Resources

The following X resources are used to coordinate the Adobe ShowPS application with the Display PostScript NX software. See Display PostScript NX documentation for more information.

`dpsnxAgent`	Selects a specific Display PostScript NX agent with which to connect. When this resource is not set, the Display PostScript NX agent default is used. Format for specifying the agent is [*hostname*]`:[:]`*portnumber.* Default is *not set*.
`dpsnxAgentExec`	Sets the path to an executable Display PostScript NX agent for autolaunching. Default value is `/usr/bin/X11/dpsnx.agent`.
`dpsnxDebugLevel`	Sets the debug level for the Display PostScript NX agent. Format for the level is an integer `0` or greater. Default is `0`.
`dpsnxExtraStack`	Adds a specified number (expressed in bytes) of extra stack space to each context created in the Display PostScript NX agent. Default is `0`.
`dpsnxLingerTime`	Sets the linger time, in hours and minutes (`hh:mm`), for the NX agent. This is the time the agent keeps listening for new connections after the last client disconnects. Default is `00:05` (five minutes).
`dpsnxNoLog`	When `True`, causes the Display PostScript NX agent to send debug log information to *`stderr`* instead of to the debug log file. Default is `False`.
`dpsnxPSResDir`	Sets path to the Display PostScript NX agent resource directory. Typical value is `/usr/lib/X11/DPS`. When this resource is not set, the Display PostScript NX agent default is used. Default is *not set*.
`dpsnxPixmapMemory`	Informs the Display PostScript NX agent about the availability of pixmap storage on the X server. One of three arguments can be used: `limited`, `moderate`, or `unlimited`. When this resource is not set, the Display PostScript NX agent default is used. Default is *not set*.
`dpsnxPortNumber`	Sets the port number for a new auto-launched Display PostScript NX agent. When this resource is not set, the Display PostScript NX agent default is used. Default is *not set*.
`dpsnxQuantum`	Sets the operator quantum for the Display PostScript NX agent. Default is `1000`.
`dpsnxSynchronous`	Sets synchronous mode in the Display PostScript Client Library. Default is `False`.
`dpsnxTransport`	Sets the transport protocol to be used by a newly auto-launched Display PostScript NX agent. Values are limited to one of the following: `tcp`, `unix`, or `decnet`. If this option is not set, the Display PostScript Client Library selects the most efficient transport. Default is *not set*.
`dpsnxUseXFonts`	Configures the Display PostScript NX agent to use X11 fonts instead of Type1 PostScript fonts whenever possible. Using this option will speed up font rendering at the cost of potentially inaccurate spacing. Default is `False`.

Chapter 4: *The filterps utility*

This chapter describes the *filterps* conversion utility. You can use *filterps* to convert incorrectly structured PostScript language documents for use with a variety of programs, including the Adobe ShowPS previewer and the TranScript™ package.

Some applications generate PostScript language documents that don't strictly adhere to the Document Structuring Conventions (DSC) described in Appendix G of the *PostScript Language Reference Manual, Second Edition*. The conventions use PostScript language comments to provide information about the structure of the document. For instance, they let an application specify where the PostScript language code for each page begins and ends.

Since the previewer expects a document that strictly adheres to the Document Structuring Conventions, it may not be able to display a document with invalid DSC comments, or may display it as an unstructured document, even if the document can be printed on a PostScript printer. Unstructured documents can't take full advantage of Adobe ShowPS features.

The *filterps* utility corrects the incorrect structure of some documents. Run it from the command line, giving the input filename and, optionally, the output filename as arguments:

```
filterps oldfile [newfile]
```

■ If you issue the command with the input filename only, *filterps* replaces *oldfile* with *newfile*.

■ If you specify `newfile` as an arguments, *filterps* converts `oldfile` and writes to `newfile`. The original file remains intact.

The program goes through the following steps:

1. It determines the application that generated the file, and checks whether a conversion routine exists for files generated by that application.
2. If no conversion routine exists, it prints a message to the terminal window.
3. If a conversion routine exists, it prints information about the conversion that is taking place to the terminal window, converts the file to a valid PostScript language file, and writes the corrected file.

Examples

In the following examples, the first line shows what you should type at the command line prompt. The second line shows what the program will print.

■ To convert a FrameMaker® 3.0 file, */uguide/doc.ps*, and overwrite the existing file:

```
%filterps /uguide/doc.ps

Applying the fixFM3.0 filter to /uguide/doc.ps, output in
/uguide/doc.ps... Done
```

■ To convert a FrameMaker 2.0 file, *olduserguide/doc.ps*, and direct the output to *fixdoc.ps*

```
%filterps /olduserguide/doc.ps fixdoc.ps

Applying the fixFM2.0 filter to /olduserguide/doc.ps, output
in /olduserguide/fixdoc.ps... Done
```

Available filters

You can invoke *filterps* with the `-help` option to get a summary of the command line syntax and a listing of available filters.

```
filterps [-help]
```

The *filterps* utility can currently filter documents generated by the following programs:

■ FrameMaker publishing software from Frame Technology Corporation, version 2.0 and version 3.0.

■ Asterix™, a collection of document creation applications from Applix™, Inc.

■ ditroff (device independent troff), a troff-like utility that is sold and licensed by AT&T Information Systems, either alone or in the Documenter's Workbench package.

■ dvi2ps, a utility that converts TeX™ documents into PostScript language documents.

Appendix A: *Troubleshooting*

This chapter explains some of the error and warning messages you may encounter while working with the Adobe ShowPS previewer. Where possible, a solution to the problem is included.

The previewer cannot load a file.

Problem: An error message informs you that the previewer cannot load a file.

Cause: The error message usually explains why the previewer cannot load the file. For example, it may tell you that the file cannot be found, that you don't have permission to load the file, or that the file does not conform to the Document Structuring Conventions.

Solution: The solution depends on why the previewer cannot load the file:

■ If the file does not conform to the document structuring conventions, you may be able to convert it with the *filterps* utility (see Chapter 4). If no filter exists for the file in question, load it as an unstructured document using the Ignore Comments menu, the `-igc` command line option, or the `ignoreComments` X resource.

■ If the `substituteFonts` X resource has been set to False, the previewer cannot load files that require unavailable fonts. Use the `-sf` option or the Substitute Fonts menu command to substitute Courier for unavailable fonts.

Commands on the Page menu are grayed or changed

Problem: You cannot bring up the Page Selection panel, and the Previous Page command has changed to First Page.

Cause: You are viewing an unstructured document (see Appendix B) or an incorrectly structured document, and the previewer therefore has no information available to allow you to select pages.

Solution: If you believe the document is actually a structured document and contains information about page selection, use the *filterps* utility described in Chapter 4. The utility converts documents generated by a number of applications to correctly structured documents. If no filter exists, you can traverse the document using the Next Page and First Page commands.

The previewer appears to display the incorrect page number.

Problem: You want to display a certain page in a document, but the previewer seems to add pages.

Cause: When the previewer chooses a page, it considers the position of the page in relation to the document's first page, not the page number in the chapter.

Solution: Take into account the page number of the first page and consider front matter such as a table of contents when specifying a page number.

A warning about pixmap buffer memory is displayed.

Problem: When you change the scale factor or use a page size larger than the default, a warning message informs you that the memory required for the pixmap has exceeded the upper limit.

Cause: By default, the previewer writes each page to an offscreen buffer called a pixmap before it displays it. The upper limit for memory that the previewer is allowed to allocate for this pixmap is set by default to 4 Mbytes. If you exceed that limit, either because you choose a larger scale factor or because you are viewing a larger page, the error message is displayed.

Solution: If you are viewing a structured document, confirm that you have seen the message by clicking on Ok. The document will then be written directly to the window. If you are viewing an unstructured document, first scroll to the area of the page you want to view. Then reopen the document by choosing Reopen from the File menu and page forward to the place you want to see.

You can also decide to use a smaller scale, or to change the memory available for the pixmap by using the `-maxp` option or the `maxPixmapBytes` X resource.

Additional Information: A large page size, whether requested by the `-ps` option or by a command in the PostScript language document, requires allocation of a substantial amount of memory. For example, on a machine that displays 100 dots per linear inch, each square inch has 10 000 dots. If the machine is an 8-bit color machine, 10 kilobytes are needed for each square inch of the page. On such a machine, the X server allocates about 16 megabytes for an A0 size page.

At times the pixmap storage required to display a page at a given scale exceeds the number specified by the `-maxp` option. In that case, the previewer displays a warning message and draws directly to the window instead of allocating a pixmap. Whether this message is displayed again depends on the action you take:

- If you reduce the scale factor but the required pixmap storage still exceeds the number specified by the `-maxp` option, the message is displayed again. This helps you determine how far the scale factor has to be reduced.
- If you increase the scale factor, the message is not redisplayed, since you already know there is not enough space. This prevents the message from popping up repeatedly as you move the scale slider.
- If you open another file without reducing the scale factor, the pages of that file cannot be written to a pixmap either. Since you have been informed already, no message is displayed.

Appendix B: *Document Considerations*

This appendix describes how the Adobe ShowPS previewer deals with different kinds of documents.

Document requirements

The previewer can display only PostScript language documents.

■ If a file contains incorrect PostScript language, the previewer may not be able to load it and will display an error dialog.

■ The file type must be indicated on the first line of the document. Otherwise, the previewer cannot open the file. Even a hand-coded PostScript language file must contain a file type indicator on the first line. Use a percent sign directly followed by an exclamation mark to indicate an unstructured document. This exact string—and nothing else—must be the first line of the file.

■ Many word processing or graphics applications let you save documents as PostScript language files, usually with the extension *.ps* or *.PS*. The previewer is designed to show these files. Don't try to preview the application files themselves.

Structured and unstructured documents

Adobe ShowPS allows you to preview structured and unstructured documents. A structured PostScript language document contains Document Structuring Convention (DSC) comments, defined in Appendix G of the *PostScript Language Reference Manual, Second Edition*. The previewer needs this information, for example, to let you select individual pages.

■ When you view a structured document, you can page forward and backward and select individual pages.

■ When you view an unstructured document, you can only page forward and return to the first page.

■ The previewer displays a trailing blank page at the end of each unstructured document.

■ If a document uses DSC comments incorrectly, the previewer may not be able to display the document, or may display it as an unstructured document. The `-igc` command-line option or the Ignore Comments menu command let you preview the document as an unstructured document. You can use the *filterps* utility to convert some incorrectly structured documents to correctly structured documents (see Chapter 4).

Special kinds of documents

The previewer provides the following services for special cases:

■ The previewer detects structured documents whose pages have been placed in reverse order and displays the pages in normal order.

■ When you view a standalone EPS file, the program displays the image with its top left corner in the top left corner of the window. That way, you can immediately view the document without having to scroll through parts of the page. To see the actual placement on the page, use the `repositionEPSF` resource, the `-repsf` command-line option, or the Reposition EPS Files command in the Options menu.

Appendix C: *Standard page sizes*

Page size options include convenient names such as “letter” for 8.5 × 11 inch sizes, as well as formal American ANSI and metric ISO designations. This appendix summarizes these size options.

In page size options, case is significant. For example, while “letter” produces 8.5 × 11 inch pages (portrait orientation), “Letter” produces pages 11 × 8.5 inches (landscape orientation).

Page sizes are not limited to those listed in the following tables. A page of arbitrary size can be displayed by specifying an argument of the form *width*:*height.* The default units for each dimension are inches, but users can also express values in units of pixels, millimeters, centimeters, or points by specifying a suffix to the values. The full syntax of the *width*:*height* argument is therefore:

width`[px|mm|cm|pt]`:*height*`[px|mm|cm|pt]`

The entries in the “Argument” columns of the tables below are to be specified as arguments to the `-ps` option. For example,

```
%showps -ps a4
```

produces a page 210 × 297 mm.

American ANSI sizes

Argument	*Width × Height (in)*	*Width × Height (mm)*
a	8.5 × 11.0	215.9 × 279.4
b	11.0 × 17.0	279.4 × 431.8
c	17.0 × 22.0	431.8 × 558.8
d	22.0 × 34.0	558.8 × 863.6
e	34.0 × 44.0	863.6 × 1117.6
f	28.0 × 40.0	711.2 × 1016.0
e1	44.0 × 68.0	1117.6 × 1727.2

The following arguments are defined for convenience in specifying popular American sizes.

Convenient arguments for American page sizes

Argument	*Width × Height (in)*	*Comment*
executive	7.25 × 10.5	
letter lettersmall lettertray note	8.5 × 11.0	Same dimensions as “a”
Letter	11.0 × 8.5	Landscape version of “letter”
legal legaltray	8.5 × 14.0	
Legal	14.0 × 8.5	Landscape version of “legal”
tabloid 11x17 11x17tray	11.0 × 17.0	Same dimensions as “b”
ledger ledgertray 17x11	17.0 × 11.0	Landscape version of “tabloid”
statement	5.5 × 8.5	
folio	8.5 × 13.0	
quarto	8.47 × 10.83	
10x14	10.0 × 14.0	

Metric ISO A sizes

Argument	*Width × Height (mm)*	*Width × Height (in)*
a0	841.0 × 1189.0	33.11 × 46.81
a1	594.0 × 841.0	23.39 × 33.11
a2	420.0 × 594.0	16.54 × 23.39
a3	297.0 × 420.0	11.69 × 16.54
a4	210.0 × 297.0	8.27 × 11.69
a5	148.0 × 210.0	5.83 × 8.27
a6	105.0 × 148.0	4.13 × 5.83
a7	74.0 × 105.0	2.91 × 4.13
a8	52.0 × 74.0	2.05 × 2.91
a9	37.0 × 52.0	1.46 × 2.05
a10	26.0 × 37.0	1.02 × 1.46

Metric ISO B sizes

Argument	*Width × Height (mm)*	*Width × Height (in)*
b0	1000.0 × 1414.0	39.37 × 55.67
b1	707.0 × 1000.0	27.83 × 39.37
b2	500.0 × 707.0	19.68 × 27.83
b3	353.0 × 500.0	13.90 × 19.68
b4	250.0 × 353.0	9.84 × 13.90

Metric ISO B sizes *(continued)*

Argument	*Width × Height (mm)*	*Width × Height (in)*
b5	176.0 × 250.0	6.93 × 9.84
b6	125.0 × 176.0	4.92 × 6.93
b7	88.0 × 125.0	3.46 × 4.92
b8	62.0 × 88.0	2.44 × 3.46
b9	44.0 × 62.0	1.73 × 2.44
b10	31.0 × 44.0	1.22 × 1.73

Metric ISO C sizes

Argument	*Width × Height (mm)*	*Width × Height (in)*
c0	914.4 × 1300.5	36.00 × 51.20
c1	650.2 × 914.4	25.60 × 36.00
c2	457.2 × 650.2	18.00 × 25.60
c3	325.1 × 457.2	12.80 × 18.00
c4	228.6 × 325.1	9.00 × 12.80
c5	162.6 × 228.6	6.40 × 9.00
c6	114.3 × 162.6	4.50 × 6.40
c7	81.3 × 114.3	3.20 × 4.50

The following arguments are defined for convenience in specifying popular metric sizes.

Convenient arguments for metric page sizes

Argument	*Width × Height (mm)*	*Comment*
a4small a4tray	210.0 × 297.0	Same dimensions as “a4”
A4	297.0 × 210.0	Landscape version of “a4”
a3tray	297.0 × 420.0	Same dimensions as “a3”
A3	420.0 × 297.0	Landscape version of “a3”
A5	210.0 × 148.0	Landscape version of “a5”
B4	353.0 × 250.0	Landscape version of “b4”
b5tray	176.0 × 250.0	Same dimensions as “b5”
B5	250.0 × 176.0	Landscape version of “b5”

Appendix D: *ShowPS Color Handling*

This appendix describes how the Adobe ShowPS PostScript document viewer coordinates with the Display PostScript system and the X window system to display colors.

Readers interested in more information on using the Display PostScript system with X should consult the following references:

1. *The Definitive Guides to the X Window System: Vol. 1: Xlib Programming Manual for Version 11* by Adrian Nye, O'Reilly & Associates, Inc. pubs., 1990. See especially Chapter 7, "Color."

2. *Programming the Display PostScript System with X*, Addison Wesley pubs., 1993.

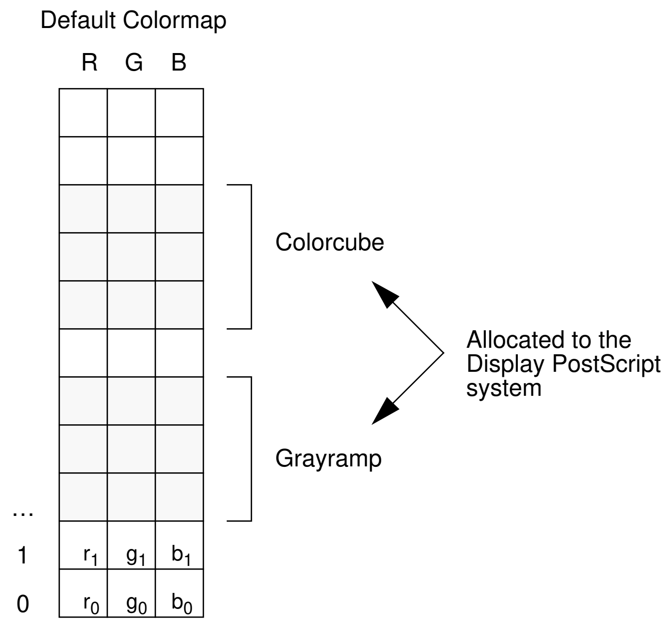
3. *Display PostScript Developer Tech Notes*, from Adobe Systems. See especially Developer Tech Note 1-001, "Level 2 Changes for X," section 4, "Creating standard colormaps."

Colormaps, colorcubes, and grayramps

The X Window System displays colors on the screen by consulting a system *colormap*. Each entry in the colormap contains three values, one for each of red, green, and blue (RGB). If the system is grayscale or monochrome, each entry in the colormap contains one value, the gray value for the pixel. The number of bits for each RGB value is the system's visual *depth*. To illuminate a color pixel on the screen, the system looks up the RGB values for the pixel in the colormap. The specific algorithm or data structure used to look up the RGB value is called the *visual class*, or *visual*.

Most graphic workstations allow one or more *virtual colormaps*, which are created in memory and installed into the system colormap at the direction of the window manager. The characteristics of a virtual colormap that has been or can be created for a particular screen are described by its visual.

The Display PostScript system, by default, claims a portion of the standard colormap for use with Display PostScript applications. The portion used to display colors is called the *colorcube*; the portion used to display shades of gray is called the *grayramp*.



Using the color-handling options

You can change the way ShowPS allocates and uses colors by using options and X resources to specify the following:

- Whether or not ShowPS is allowed to create a separate colormap when the default system colormap is not large enough to allocate the requested colorcube and grayramp.
- Whether or not ShowPS releases the colormap space it has reserved for its colorcube and grayramp when it exits.
- The visual depth.
- The visual class.
- The preferred number of reds, greens, blues, and grays reserved for ShowPS in the colormap.

The options and X resources used to control ShowPS color-handling are summarized in the table on page 56.

Allowing a separate colormap

By default, ShowPS allocates its colorcube and grayramp in the default system colormap. The `-sepcmap` option and its associated `colormapAllowSeparateMap` X resource let you direct ShowPS to create a separate (virtual) colormap if there is not enough room in the default colormap. The following scenarios illustrate cases in which you might want to use a separate colormap.

First, consider the case in which you have opened several color-intensive applications that require most of the colors in the colormap. If you then launch ShowPS, it may not be able to allocate enough colors to display a color document. In the worst case, the document will appear in monochrome (black and white).

The problem can be remedied with the `-sepcmap` option. By launching ShowPS with the `-sepcmap` option (or setting your `colormapAllowSeparateMap` X resource to `True`), you allow ShowPS to create a separate colormap if it cannot allocate enough colors in the default system colormap. ShowPS will then display the document in the best colors allowed by the selected visual class and visual depth. A disadvantage for some systems is that other applications may be shown in false colors when the window manager's focus is on ShowPS. The false colors may cause screen flashing as you shift user focus (and colormaps) between ShowPS and other application windows.

Some advanced systems are capable of installing several colormaps simultaneously. Users typically observe little or no screen flashing on these systems as they shift focus between ShowPS and other application windows. If you have such a system, you may want to set the `colormapAllowSeparateMap` resource to `True` permanently.

Retaining colors in the colormap

The default behavior of ShowPS is to release the colors it has allocated in the color map when it exits. Releasing the colors allows other applications to use all the colors in the colormap as required. The only disadvantage is that other Display PostScript applications (other ShowPS processes, for example) cannot share the same color cube and gray ramp, and might exhaust the colormap. If you are planning to run several Display PostScript applications simultaneously, use the `-retain` option (or set the `colormapRetainColors` resource to `True`) when starting ShowPS.

Note that you can only retain colors in the default colormap. The `-retain` option does not apply in cases in which ShowPS creates a separate colormap, because a separate colormap that was created for an instance of ShowPS disappears when the ShowPS process terminates.

Releasing colors in the colormap

Many applications create and allocate standard colormaps in the default system colormap. These applications typically allocate the colormaps in *retained* mode, and post the properties RGB_DEFAULT_MAP and DEFAULT_GRAY on the root window to allow other applications to use and share the standard colormaps. When you start ShowPS with the `-retain` option, colors for ShowPS are allocated in retained mode.

When colors are allocated in retained mode, the standard colormaps and allocated colorcells remain allocated in the default system colormap after the application exits and until the X server is restarted. The disadvantage of retained mode may become evident when you run other color-intensive applications that do not use the standard colormap properties. These applications are unable to use all the available colors in the colormap because some of the colors are already allocated for the standard colormaps.

The `killstdcmap`(1) utility enables you to release the colorcells associated with the RGB_DEFAULT_MAP and DEFAULT_GRAY properties. It will also remove the properties from the root window so other applications don't assume that the colors are available. If you want to remove the RGB_DEFAULT_MAP and DEFAULT_GRAY properties from your root window, but the `killstdcmap` utility is not available on your system, you can instead remove the properties manually with the following steps:

1. Run the following commands and make note of the kill id number for each property:

```
xprop -root RGB_DEFAULT_MAP
xprop -root DEFAULT_GRAY
```

2. Release the colorcells by running the following commands on each kill id

```
xkill -id kill_id
```

3. Run the following commands to remove the properties from the root window:

```
xprop -root -remove RGB_DEFAULT_MAP
xprop -root -remove DEFAULT_GRAY
```

If the above commands succeed, the standard colormap properties and the associated colorcells are removed from your system's default colormap.

> *CAUTION: Please be aware that running `killstdcmap` will remove the RGB_DEFAULT_MAP and DEFAULT_GRAY properties on the root window and release the associated colorcells, even if other applications are using them. If you have not exited those other applications before running `killstdcmap`, those applications might display in false colors, even when the window manager pointer focus is moved into the application windows. The reason is that, by convention, applications can use the standard colormap properties that are advertised on the root window and do not need to allocate the colorcells. `killstdcmap` therefore cannot tell if the properties are being used by other applications. This is the reason that `killstdcmap` is a separate utility under user control and not a built-in feature of an application.*

Specifying the size of the DPS colorcube and grayramp

The resources `colormap[Reds|Greens|Blues|Grays]` (and corresponding command-line options) allow you to specify how many values of each color to use for constructing the Display PostScript colorcube and grayramp. As described in *Programming the Display PostScript System with X*, section 3.2.1 "Using **XDPSCreateSimpleContext**," specifying larger values improves image quality by allowing more pure colors and requiring less halftoning.

Specifying zero as a color value

If one of these resources (or option values) is set to zero, then ShowPS uses resources in your `.Xdefaults` file to determine the value for that resource. You can configure this usage by setting resource values in your `.Xdefaults` file. Giving ShowPS more colormap entries improves the quality of its rendering but leaves fewer entries available to other applications if the colormap is the default system colormap.

Each resource entry in your `.Xdefaults` file should be of the form

`DPSColorCube.`*`visualType`*`.`*`depth`*`.`*`color`*`: `*`size`*

where

`visualType` is one of `GrayScale`, `PseudoColor`, or `DirectColor`.

`depth` is 1, 2, 4, 8, 12, or 24.

`color` is one of the strings "reds", "greens", "blues", or "grays".

`size` is the number of values of that color to allocate.

These resources are not used for the static visual types `StaticGray`, `StaticColor`, or `TrueColor`.

Specifying 0 for `reds` directs ShowPS to use only a grayramp. This specification is particularly useful for grayscale systems that incorrectly use `PseudoColor` as the default visual.

For example, to configure a 5×5×4 color cube and a 17-element grayramp for an 8-bit `PseudoColor` screen, specify these resources:

```
DPSColorCube.PseudoColor.8.reds: 5
DPSColorCube.PseudoColor.8.greens: 5
DPSColorCube.PseudoColor.8.blues: 4
DPSColorCube.PseudoColor.8.grays: 17
```

These resources use 117 colormap entries, 100 for the colorcube and 17 for the grayramp. For the best rendering results, specify an odd number for the grayramp. An odd number ensures that a 50% gray can be rendered without halftoning (dithering).

Resources that are not specified take these default values:

```
DPSColorCube.GrayScale.4.grays: 9
DPSColorCube.GrayScale.8.grays: 17

DPSColorCube.PseudoColor.4.reds: 2
DPSColorCube.PseudoColor.4.greens: 2
DPSColorCube.PseudoColor.4.blues: 2
DPSColorCube.PseudoColor.4.grays: 2
DPSColorCube.PseudoColor.8.reds: 4
DPSColorCube.PseudoColor.8.greens: 4
DPSColorCube.PseudoColor.8.blues: 4
DPSColorCube.PseudoColor.8.grays: 9
DPSColorCube.PseudoColor.12.reds: 6
DPSColorCube.PseudoColor.12.greens: 6
DPSColorCube.PseudoColor.12.blues: 5
DPSColorCube.PseudoColor.12.grays: 17

DPSColorCube.DirectColor.12.reds: 6
DPSColorCube.DirectColor.12.greens: 6
DPSColorCube.DirectColor.12.blues: 6
DPSColorCube.DirectColor.12.grays: 6
DPSColorCube.DirectColor.24.reds: 7
DPSColorCube.DirectColor.24.greens: 7
DPSColorCube.DirectColor.24.blues: 7
DPSColorCube.DirectColor.24.grays: 7
```

If none of the above defaults apply to the display, ShowPS uses no colorcube and a 2-element gray ramp; that is, black and white.

ShowPS uses the Display PostScript Client Library procedure **XDPSCreateStandardColormaps** for allocating its colors. Interested readers can consult Developer Tech Note 1-001, “Level 2 Changes for X,” section 4, “Creating standard colormaps,” for a detailed description.

Specifying a visual class and depth

The following scenarios illustrate why you might want to specify a particular visual class and visual depth with the `visualClass` and `visualDepth` X resources and their associated options.

Eliminating halftoning on 12- and 24-bit systems

On most systems, the default visual has a depth of 8 bits, which may require ShowPS to use halftoning for some colors. For the best possible display, you may want to display color pictures without halftoning. If your system has a 12- or 24-bit `TrueColor` visual, you can specify it to eliminate halftoning. Use the `visualClass` and `visualDepth` resources and their associated options to specify the visual.

Conserving system memory

Assume that you want to use the pixmap (offscreen buffer) to maintain the best system performance (see `-maxp` option on page 26), but limit the color depth to 12 bits in order to conserve system memory. Given those constraints, assume that you are content to let ShowPS find the best visual. To apply these constraints, you would set the `visualClass` resource to `Best` and the `visualDepth` resource to `12`. ShowPS would then find the best visual that was not deeper than 12 bits.

Setting a specific visual and depth

If you are an advanced user, you might want to set a specific visual and depth. For example, assume that you want to set up an experiment using `PseudoColor` at depth 12. In that case, you would set `visualClass` to `PseudoColor` and `visualDepth` to `12`. ShowPS would then use `PseudoColor` at a depth of 12 bits. If `PseudoColor` were not available at 12 bits, ShowPS would search the available visuals for `PseudoColor` at successively lower depths until it found one.

Color-handling options

The following table summarizes the ShowPS X resources and command-line options that enable you to control system color usage.

X resource	Command-line option	Type, values, and defaults	Description
`colormapRetainColors`	`-retain`	Type: `boolean` Default: `False`	If this resource is set to True and ShowPS has allocated colors in the default colormap, those colors will persist (remain allocated in the colormap) after ShowPS exits. If the visual is not the default visual, or if ShowPS creates a separate colormap, then the allocated colors will *not* be retained, regardless of the value of this resource or option.
`colormapAllowSepa-rateMap`	`-sepcmap`	Type: `boolean` Default: `False`	Allows ShowPS to create a separate colormap if the Display PostScript colorcube and grayramp do not fit in the default colormap.
`colormapReds`	`-reds` *number*	Type: `integer` Values: `0`, `2–256` Default: `0`	Sets the preferred number of red values to use in the Display PostScript colorcube.
`colormapGreens`	`-greens` *number*	Type: `integer` Values: `0`, `2–256` Default: `0`	Sets the preferred number of green values to use in the Display PostScript colorcube.
`colormapBlues`	`-blues` *number*	Type: `integer` Values: `0`, `2–256` Default: `0`	Sets the preferred number of blue values to use in the Display PostScript colorcube.
`colormapGrays`	`-grays` *number*	Type: `integer` Values: `0`, `2–256` Default: `0`	Sets the preferred number of gray values to use in the Display PostScript grayramp.
`visualClass`	`-visual` *visual*	Type: `string` Values: `Best`, `Default`, `StaticGray`, `GrayScale`, `StaticColor`, `PseudoColor`, `TrueColor`, `DirectColor` Default: `Default`	Sets the visual class for ShowPS to use.
`visualDepth`	`-depth` *depth*	Type: `integer` Values: `1`, `2`, `4`, `8`, `12`, `24` Default: 24	Sets the preferred visual depth for ShowPS to use.

How ShowPS color resources and options work together

Here is the algorithm that ShowPS follows when setting up the visual, the colormap, and the Display PostScript colorcube and grayramp. The logic of visual and colormap selection is presented in pseudocode format.

1. Validating resources and options

During start-up, ShowPS reads and validates the command-line options and resources to make sure they contain legal values.

2. Selecting a visual

ShowPS then selects a visual according to the following pseudocode:

---

if the user selected `Default` visual, **then**
do
    Select the default visual and ignore the preferred visual depth.
end
else
if the user selected `Best` visual, **then**
do
    For each depth, starting at the preferred depth specified by the `visualDepth` resource, look at the available visuals at that depth, and select the "best" visual. If no visuals are available at the current depth, go to the next lower depth. The search order for the "best" visual is different for each depth, as follows:

24 bit:	`TrueColor,`	`DirectColor,`	`PseudoColor,`
	`StaticColor,`	`GrayScale,`	`StaticGray`
12 bit:	`TrueColor,`	`PseudoColor,`	`DirectColor,`
	`StaticColor,`	`GrayScale,`	`StaticGray`
8 bit:	`PseudoColor,`	`TrueColor,`	`StaticColor,`
	`GrayScale,`	`StaticGray,`	`DirectColor`
4 bit:	`PseudoColor,`	`TrueColor,`	`StaticColor,`
	`GrayScale,`	`StaticGray,`	`DirectColor`
2 bit:	`PseudoColor,`	`StaticColor,`	`GrayScale,`
	`StaticGray,`	`TrueColor,`	`DirectColor`
1 bit:	`StaticGray,`	`GrayScale,`	`PseudoColor,`
	`StaticColor,`	`TrueColor,`	`DirectColor`

Some of the visuals obviously do not make sense at certain depths. For example, at 24 bit depth, only `TrueColor` or `DirectColor` would normally be available; at 1 bit depth, only `StaticGray` makes sense. However, for completeness ShowPS maintains all six visuals in the search table for all depths. If no suitable visual is found at any depth (very unlikely), ShowPS exits with an error.

end
else
do

The user has requested a specific visual, for example `PseudoColor`. For each depth, starting at the preferred depth specified by the `visualDepth` resource, look at the available visuals at that depth. If the requested visual is available at the current depth, select it. Otherwise, go to the next lower depth. If the requested visual is not available at any depth, ShowPS exits with an error.

end

3. Selecting a colormap

ShowPS then selects a colormap according to the following pseudo code:

---

if the final selected visual is the default visual, **then**
do
    if the selected visual is read-only (`StaticGray`, `StaticColor`, or `TrueColor`), **then**
    do
        Override the `colormapAllowSeparateMap` resource (set it to `False`).
        Select the default colormap and create the Display PostScript colorcube and grayramp in the default colormap. In this case, the sizes of the actual colorcube and grayramp may be reduced to fit in the available space in the default colormap, according to the rules described in Display PostScript Developer Tech Note 1-001, section 4, in the description of **XDPSCreateStandardColormap**.
    end
    else
    if the `colormapAllowSeparateMap` resource is `True`, **then**
    do
        if there *is* enough room for the Display PostScript colorcube and grayramp in the default colormap, **then**
        do
            Select the default colormap and create the Display PostScript colorcube and grayramp in it.
        end
        else
        do
            Override the `colormapRetainColors` resource (set it to `False`).
            Create a new colormap and select it.
            Create the Display PostScript color cube and graymap in the new colormap.
        end
    end
    else

do

Select the default colormap and create the Display PostScript colorcube and grayramp in the default colormap. In this case, the sizes of the actual colorcube and grayramp may be reduced to fit in the available space in the default colormap, according to the rules described in Display PostScript Developer Tech Note 1-001, section 4, in the description of **XDPSCreateStandardColormap**.

end

end

else

The visual is not the default visual, so:

do

Override the `colormapRetainColors` resource (set it to `False`).
Override the `colormapAllowSeparateMap` resource (set it to `True`).
Create a new colormap and select it.
Create the Display PostScript colorcube and grayramp in the new colormap.

end

Index

E

F

G

I

K

L

M

X

Z

Adobe ShowPS Registration

Your registration will enable Adobe to keep you up to date about new product releases and other important information.

To register your copy of Adobe ShowPS, fill out this page and fax it to Adobe Systems Incorporated at 1-415-903-0589. You can also register by e-mail by sending the information requested on the card to showps@adobe.com.

Your Name: ____________________

Title: ____________________

Organization/Company: ____________________

Address: ____________________

____

____

Telephone: Fax: ____________________ Voice: ____________________

E-mail address: ____________________

On what workstation are you running ShowPS? ____________________

ShowPS version number (shown at top of ShowPS document window): ____________________