

# The Display PostScript™ System Communications Handbook



## Contents

❖	What is the Display PostScript System?	<b>3</b>
❖	Benefits of the Display PostScript System	<b>5</b>
❖	Develop with the Display PostScript System and Save Time	<b>6</b>
❖	Display PostScript Extension to X	<b>7</b>
❖	Display PostScript NX Software	<b>8</b>
❖	Accessing the Display PostScript System	<b>10</b>
❖	Building a Display PostScript Application	<b>12</b>
❖	Developer Information	<b>13</b>

---

## What Is the Adobe™ Display PostScript System?

### Advanced graphics software.

The Display PostScript system is an advanced graphics library that can be used by applications running in the X Window System™ environment. Applications use X to provide the basic windowing system and use the Display PostScript system to provide high-quality, device independent text, graphics and images in each window.

### Standard PostScript™ language imaging.

With the Display PostScript System, applications can use the PostScript language to create graphics for both the screen and the printer. The PostScript language is already a standard for printed output. As shown in Figure 1, applications can now use the same language they send to the printer to display high-quality interactive graphics.

The power and flexibility of the PostScript language make it possible to render lines with precision, rotate and scale images at will, manipulate type as a graphic object, and image virtually any two-dimensional page imaginable. Your C or C++ application can use specific imaging instructions written in the PostScript language. You may not have to write any PostScript language code—you can simply call procedures in the Display PostScript Toolkit to automatically provide PostScript imaging on the display.

### An extension to the X Window System or a client application.

The Display PostScript system has two implementations that use the same application programming interface (API): the Display PostScript extension to X and Display PostScript NX software. Many workstation manufacturers ship the Display PostScript extension to X with their X servers. X terminal manufacturers

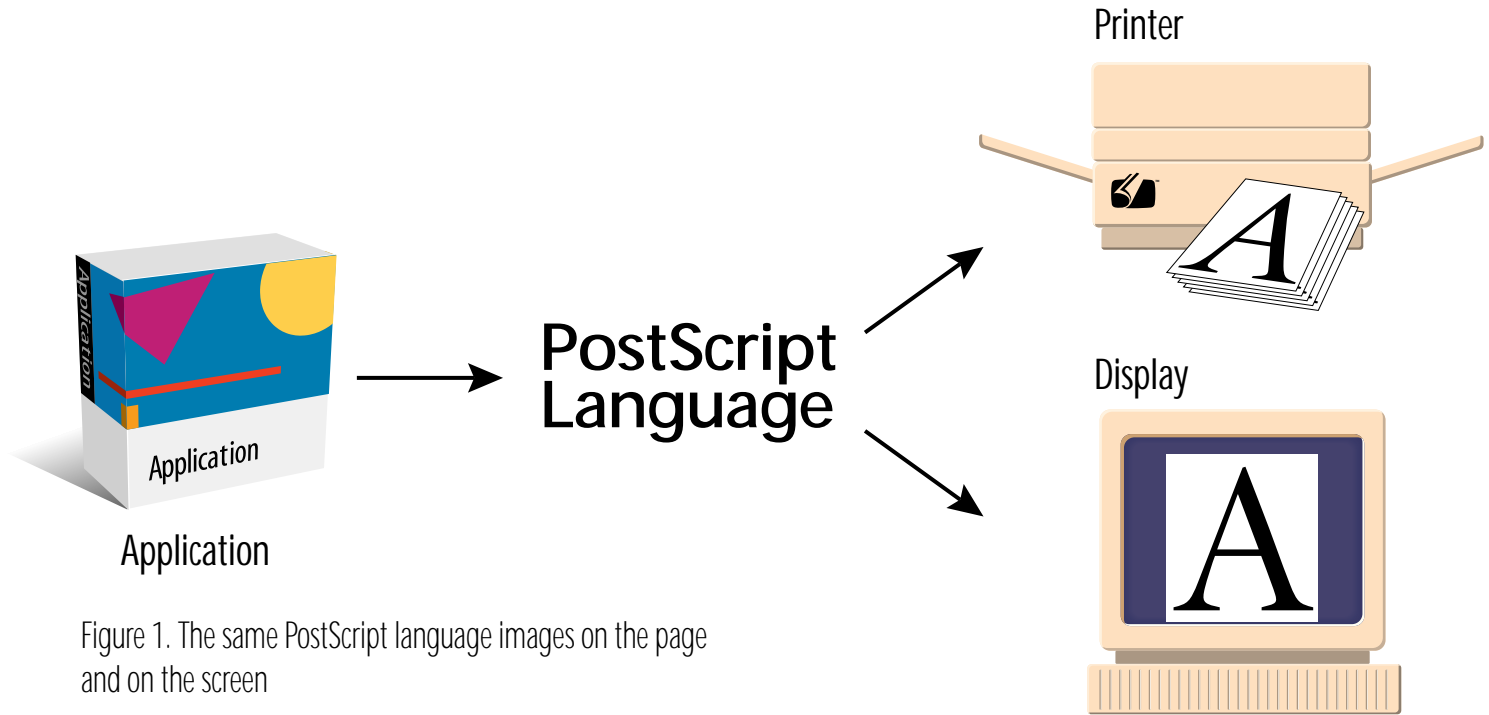


Figure 1. The same PostScript language images on the page and on the screen

applications can bundle Display PostScript NX software, a client application that supplies PostScript capability. Your application can automatically use the Display PostScript

extension, if it is present, or launch Display PostScript NX software.

---

## Benefits of the Display PostScript System

### Type 1 fonts.

The Display PostScript system uses the same outline fonts as PostScript printers: Type 1 fonts. Numbering over 2,000, the Adobe Type Library is the largest collection of high-quality fonts in electronic printing and publishing. Your application can now access this and other Type 1 font libraries. Since Type 1 fonts are created from outlines, not bitmaps, the Display PostScript system can rotate, scale and skew these fonts without compromising quality.

### Fabulous graphics.

There is no limit to the number of colors, sophisticated graphics and quality fonts available to your application. PostScript is a high-level, powerful graphics language. The limited color and graphics of X no longer limit your application. By using the

Display PostScript system, your application can have sophisticated graphics.

### Great imaging on any device.

Users can view a Display PostScript application on a 24-bit true-color, 8-bit color, grayscale or monochrome display. The Display PostScript system renders output with the best possible representation for each output device—true color on true-color devices, shades of gray on grayscale devices, and black-and-white halftones on monochrome devices. Applications that use the Display PostScript system can display on UNIX® workstations, X terminals or PCs.

### View exactly what you print—no print surprises.

Most X applications use one language to describe a page for display and another for printing. When applications have to translate page descriptions from display language to printer language, the

displayed and printed pages don't match. The unified PostScript imaging model creates consistency among displays and printers. Users of applications based on the Display PostScript system can print to a PostScript printer and be certain that paper will match screen.

---

## Develop with the Display PostScript System and Save Time

### Same code to image on all displays.

Applications with PostScript based graphics can use the same code for all display types. Supporting unanticipated display types need not require a product enhancement. The Display PostScript system can render graphics at a resolution appropriate for the display. PostScript language fonts, graphics and images will always be the highest quality that device capabilities permit.

### One imaging model for both display and printer.

Use the PostScript language to describe graphics for the screen, and then use the PostScript language to send graphics to the printer. You do not need to convert between two different imaging models and write twice as much code. Much of the code you write for the display can also be used for printing. More reusable code means decreased development costs and a shorter development cycle.

### Portable graphics code.

Graphics written in the PostScript language don't depend on the machine used to create them or the output device used to display them. Graphics code that is written in the PostScript language can be ported directly to any platform running X.

### Easy migration from X-based graphics.

X graphics and PostScript language graphics can peacefully co-exist. Existing

X applications can incrementally add Display PostScript features as they are updated or enhanced. The first step in upgrading an application could be to use the Display PostScript system to rasterize Type 1 fonts. Another frequently added feature is the ability to preview PostScript language output. Determine the advantages offered by the Display PostScript system that are most important to your installed base, and then gradually add them as enhancements are rolled into the application.

---

## Display PostScript Extension to X

### Great graphics on UNIX workstations.

The Display PostScript extension to X adds fast, high-quality text and graphics to X servers. Implemented as a standard extension to X, the Display PostScript Client Library communicates with the extension using standard X protocol. X-based applications using the Display PostScript system can run over the network like any other X application.

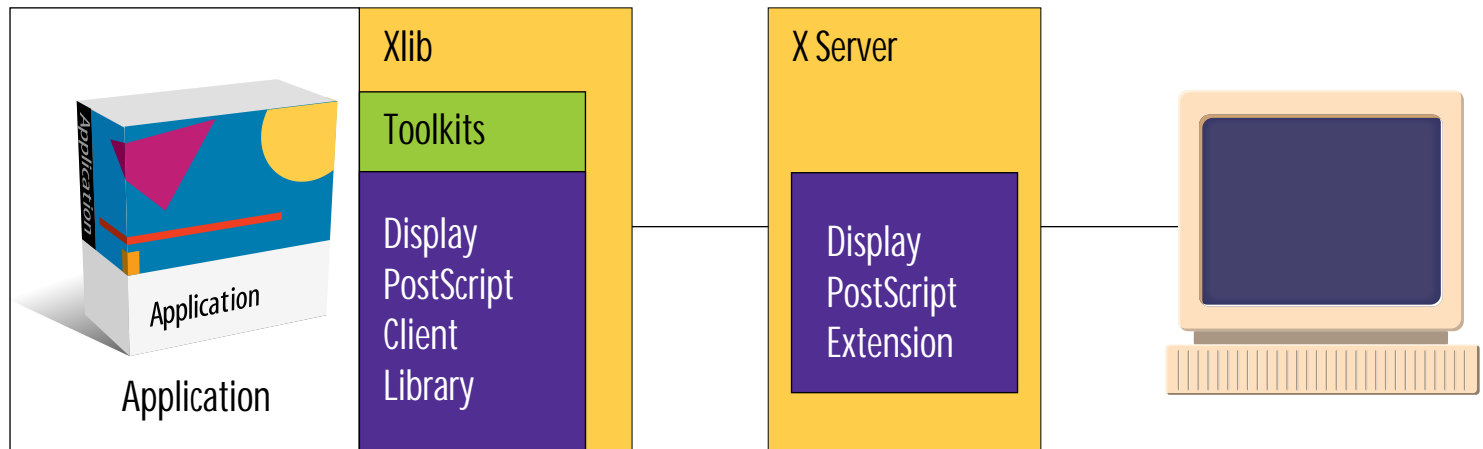


Figure 2. Communicating with the Display PostScript extension

## Architecture.

Figure 2 illustrates the information flow between the application and the X server when the Display PostScript system is implemented as an extension to X. The X server and its associated window manager handle window operations such as displaying and positioning windows, while the Display PostScript extension handles imaging within the window. The application communicates with the server by means of X requests, replies and events.

---

## Display PostScript NX Software

### Can display on any X server.

Display PostScript NX software is a host-based program that contains the PostScript interpreter. Instead of being integrated with the X Window System, the software communicates with the X server in the same way that an application does.

This arrangement lets your Display PostScript application create high-quality text and graphics on any X server—on millions of existing X displays, including X terminals, PC X servers and workstations—without the Display PostScript extension to X.

## Architecture.

As shown in Figure 3, a Display PostScript NX software installation consists of three major elements: the application, the Display PostScript NX agent and the X server. The PostScript interpreter is implemented as part of the agent, which is a separate process from the application. Because the interpreter is separate from the X server, the two can communicate over a network and do not need to reside on the same host.

The lines between the three elements represent information flow in the form of requests, replies and events. Display PostScript communications flow between

the Client Library and the agent; X communications flow between the X server and the application, and between the X server and the agent.

### Display PostScript NX agent.

The agent executes PostScript language code and converts it into X drawing primitives. These drawing primitives are sent to the X server for rendering in the application window. The agent does not

tie up the network by downloading large bitmaps; it communicates with the X server using high-performance X protocol. The agent has the characteristics of both an X server and an X client. Like an X server, it can accept network connections, process protocol requests, and multiplex simultaneous accesses. Like an X client, it sends drawing requests to the X server through Xlib.

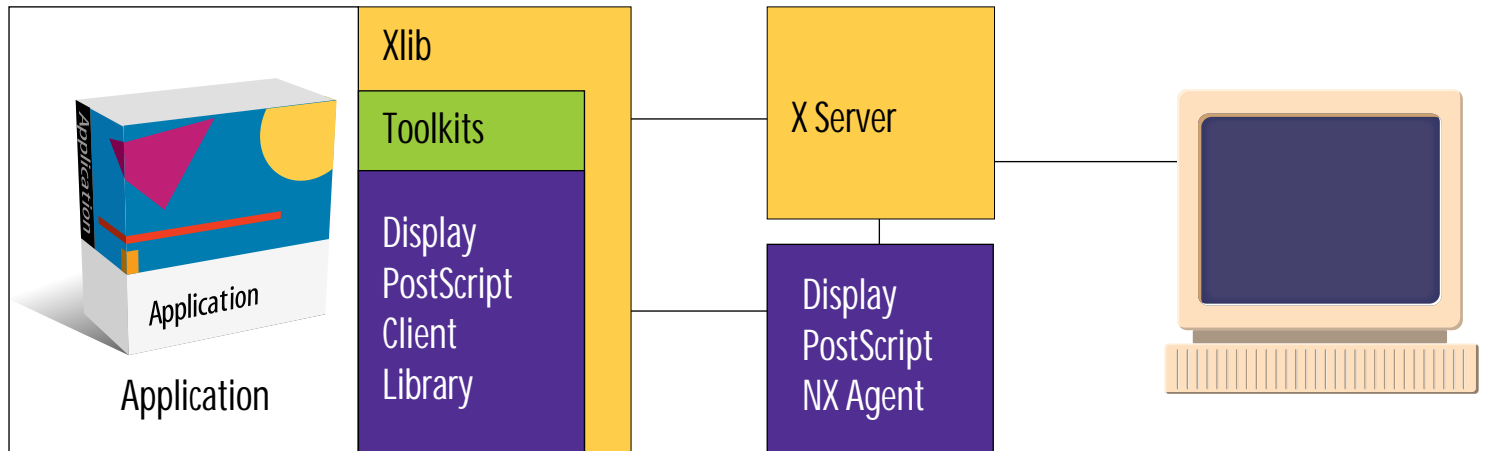


Figure 3. Communicating with Display PostScript NX software

## The agent and the X server.

The agent maintains a connection with the X server that is separate from the application's connection to the X server. Several applications can share the same agent. To guarantee that X server resources are properly preserved, the agent makes a new connection to the X server for each application. The application, agent and X server may be distributed across different hosts on a network, or they may all run on the same host.

---

## Accessing the Display PostScript System

### One set of interfaces.

By linking your application with a single set of APIs, it will work automatically with systems running either the Display PostScript extension or the Display PostScript NX software. You don't need to know if the extension or the agent is doing the work.

### Four ways to communicate with the PostScript interpreter.

Your application can use four different methods to access the Display PostScript interpreter:

Send PostScript language code as text.

The Display PostScript system provides a function call, `DPSprintf`, that can send text straight to the PostScript interpreter.

Call single-operator procedures from the Display PostScript Client Library.

Each PostScript language command is mirrored by a C language Client Library function that executes the PostScript command. Your application might use single PostScript operator functions when using the Display PostScript System as a single graphics library.

Create PostScript language procedures called wraps.

PostScript language code can be surrounded by a C language syntax and turned into a C-callable procedure using the pswrap translator. The pswrap translator enables you to describe text and graphics using the PostScript language, and then compile and link it with the C or C++ code used for the rest of the application.

Call functions in a Display PostScript toolkit.

Several toolkits have been created to accelerate the process of developing applications using the Display PostScript system. For more information, refer to the “Developer Information” section later in this booklet.



---

## Building a Display PostScript Application

Figure 4 shows how you bring together wraps, libraries, toolkits and application code to make an executable program.

### Putting it all together.

Your application source code can contain calls to wraps, toolkit routines and Display PostScript Client Library routines.

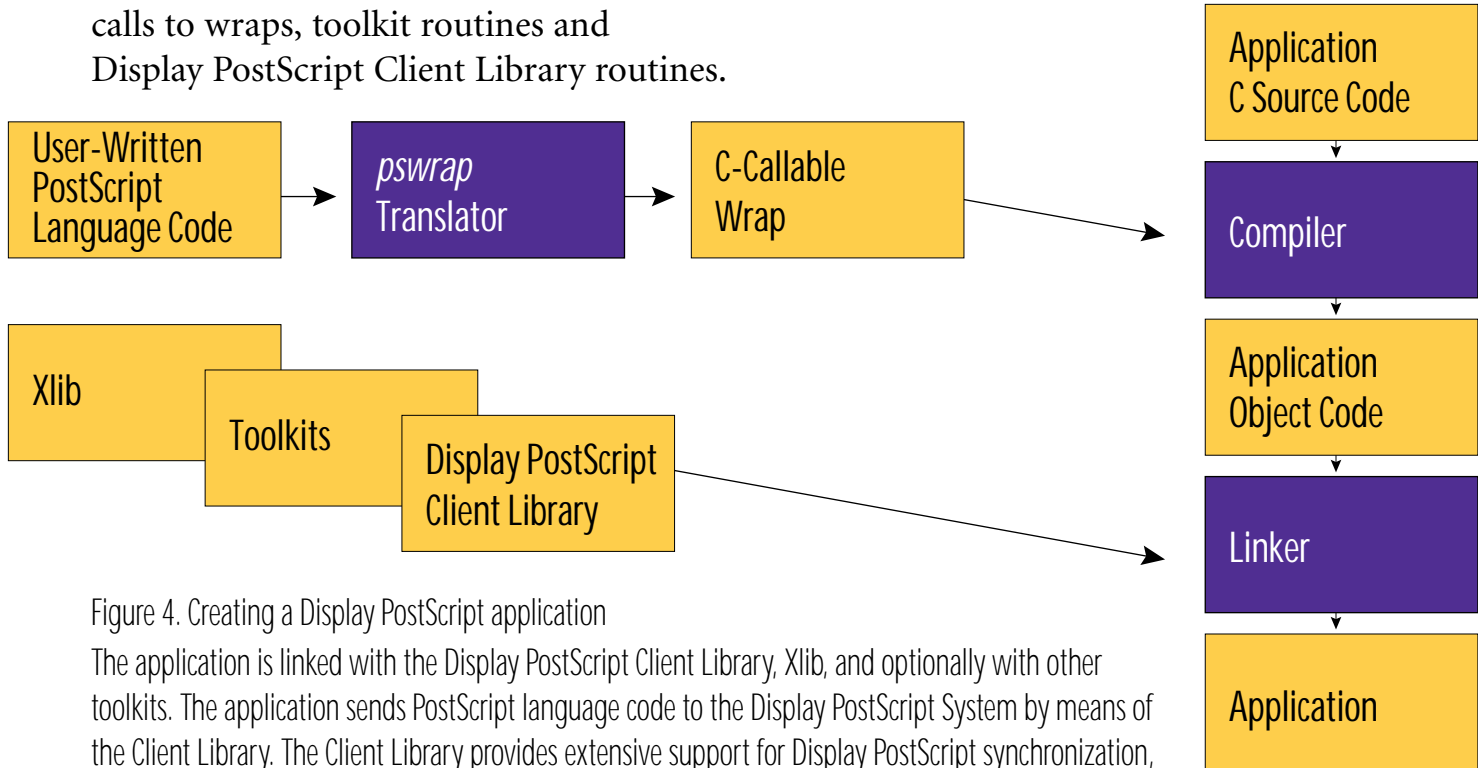


Figure 4. Creating a Display PostScript application

The application is linked with the Display PostScript Client Library, Xlib, and optionally with other toolkits. The application sends PostScript language code to the Display PostScript System by means of the Client Library. The Client Library provides extensive support for Display PostScript synchronization, request management, status management and event management. Xlib handles application communications with the X server.

---

## Developer Information

### Display PostScript Developer Information.

This packet contains the latest information about the Display PostScript system and Adobe's UNIX applications. Included is a complete list of Display PostScript extension and Display PostScript NX software distributors, training schedules, development kits and other pertinent information. To obtain a Display PostScript Developer Information kit, send e-mail with your name, company, address and phone number to [dps-info@mv.us.adobe.com](mailto:dps-info@mv.us.adobe.com).

### Display PostScript Developer Roadmap.

This packet contains information about the Adobe Developers Association (ADA), Adobe public file server, fax request line and ftp site. In addition to many other services, the ADA publishes a newsletter that covers technical issues, answers questions and announces new products.

To obtain a Display PostScript Developer Roadmap packet, send e-mail with your name, company, address and phone number to [dps-roadmap@mv.us.adobe.com](mailto:dps-roadmap@mv.us.adobe.com).

### Training.

Third-party sources offer training in the PostScript language and in developing applications using the Display PostScript system. For a list of scheduled classes, send for a Display PostScript Developer Roadmap packet.

### Toolkits.

Display PostScript application developers can leverage existing Display PostScript based widgets and utilities to expedite the development process. For information about available toolkits, send for a Display PostScript Developer Information packet.

---

Adobe, the Adobe logo, Display PostScript, PostScript and the PostScript logo are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. X Window System is a trademark of the Massachusetts Institute of Technology. All other brand or product names are trademarks or registered trademarks of their respective holders.



Copyright © 1994 Adobe Systems Incorporated.  
All rights reserved.