# ULTRIX Worksystem Software

## Display PostScript® Perspective for Software Developers

Order Number: AA–PAN7A–TE

# DISPLAY POSTSCRIPT
SYSTEM

# Perspective for Software Developers

Order No. AA–PAN7A–TE

ADOBE SYSTEMS
INCORPORATED

**Perspective for Software Developers**

Writer: Amy Davidson

October 25, 1989

The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. The software described in this document is furnished under license and may only be used or copied in accordance with the terms of such license.

This document replaces the previous version dated October 10, 1988.

# Contents

1 About This Document 1

2 About the DISPLAY POSTSCRIPT System 1

3 Application Building Blocks 2

4 Using *pswrap* 3

5 The Client Library Interface 5

6 Manuals for Application Developers 5

Index 9

# List of Figures

**Figure 1**: *The DISPLAY POSTSCRIPT System* 2

**Figure 2**: *Creating a DISPLAY POSTSCRIPT Application* 4

## 1 ABOUT THIS DOCUMENT

This document is your introduction to application development using the text and graphics imaging resources of the DISPLAY POSTSCRIPT® system.

This introduction for the application programmer describes:

- The DISPLAY POSTSCRIPT system environment you'll be interacting with.
- The Client Library interface you'll be programming for.
- The use of the *pswrap* translator to prepare C-callable procedures containing POSTSCRIPT® language programs.
- The manuals you'll need.

## 2 ABOUT THE DISPLAY POSTSCRIPT SYSTEM

The DISPLAY POSTSCRIPT system provides a device-independent imaging model for displaying information on a screen. This imaging model is fully compatible with the imaging model used in POSTSCRIPT printers. By allowing you to use the POSTSCRIPT language to display text, graphics, and sampled images, it frees you from display-specific details such as screen resolution and number of available colors.

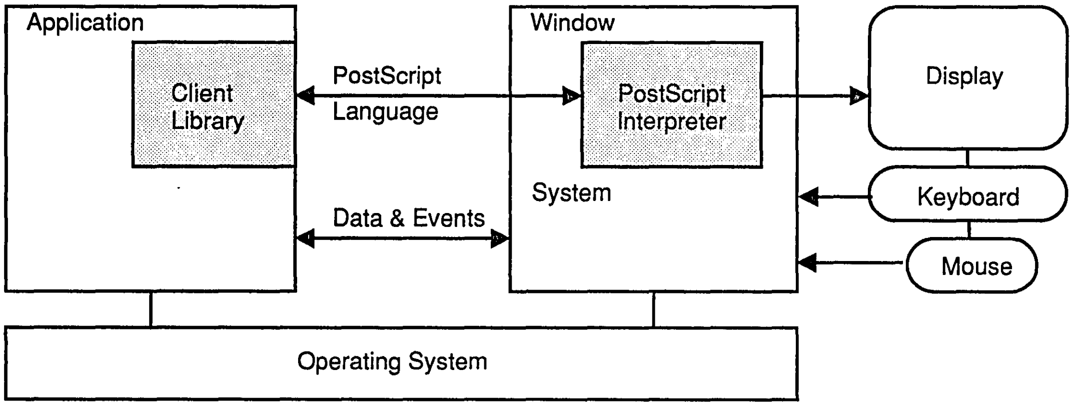
You can look at the DISPLAY POSTSCRIPT system as part of the window system. Your application will use window system features for window placement and sizing, menu creation, and event handling, while using DISPLAY POSTSCRIPT features to take care of imaging inside the window.

DISPLAY POSTSCRIPT system components include the Client Library, the POSTSCRIPT interpreter, and the *pswrap* translator. These components are described in the accompanying set of manuals. If you are a new POSTSCRIPT language programmer, you'll also need copies of the POSTSCRIPT language manuals. See Section 6 for a discussion of documentation required by application developers.

Figure 1 shows the relationship between your application and the

DISPLAY POSTSCRIPT system. The Client Library contains data structures and procedures your application will use to send requests to the POSTSCRIPT interpreter. No matter what operating system you're dealing with, you can send POSTSCRIPT language programs to the interpreter in the same way, using the same procedure calls. Therefore portability issues are minimized if you're developing an application for more than one environment.

**Figure 1** *The DISPLAY POSTSCRIPT System*



## 3 APPLICATION BUILDING BLOCKS

Most of your application will be written in C or another high-level languge. You'll call Client Library procedures to start a POSTSCRIPT execution context, send programs and data to the POSTSCRIPT interpreter, and get results from the the interpreter. The Client Library is the application's primary interface to the DISPLAY POSTSCRIPT system.

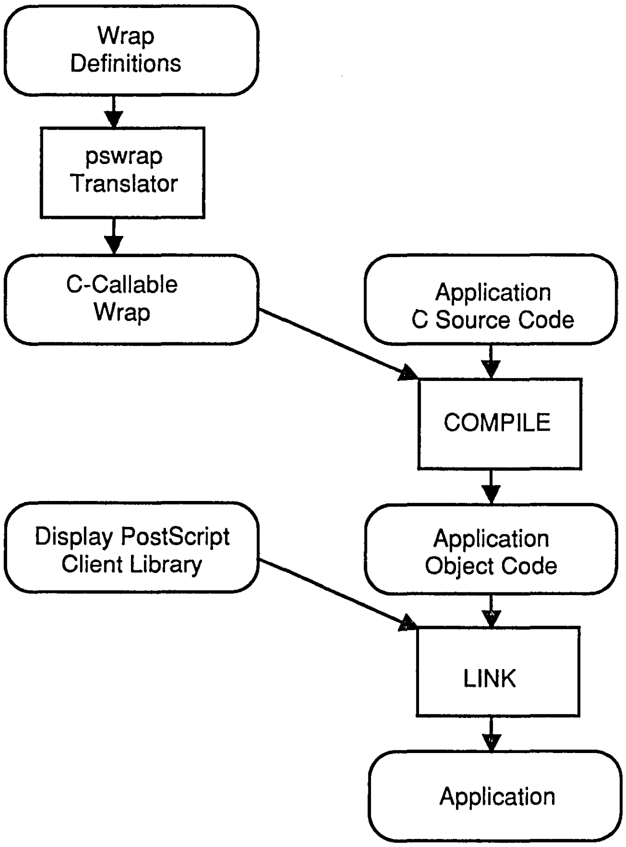
In addition, you'll call *wraps* — POSTSCRIPT language procedures developed specifically for your application. Wraps, short for *wrapped procedures*, are created by the *pswrap* translator from POSTSCRIPT language programs written to meet application needs. Figure 2 on page 4 shows how an application program is built.

## 4 USING *PSWRAP*

Your application will perform calculations, communicate with the window system, read and write files, and do other application processing in C or another high-level language. It will perform imaging tasks by calling wrapped procedures to send POSTSCRIPT language programs to the interpreter. The *pswrap* translator creates these wraps from POSTSCRIPT language input.

Figure 2 shows how an application is built of C source code and POSTSCRIPT language code translated into C-callable procedures by *pswrap*.

**Figure 2** *Creating a* DISPLAY POSTSCRIPT *Application*



Consider a wrap, *PWDisplayText*, that places text on the screen at a particular *x,y* coordinate. A call to *PWDisplayText* from the application program might look something like this:

```
PWDisplayText(72.0, 100.0, "Hello World");
```

The body of the *PWDisplayText* procedure, is actually written in the POSTSCRIPT language. It was defined to *pswrap* as follows:

```
defineps PWDisplayText(float X,Y; char *text)
    X Y moveto
    (text) show
endps
```

In the wrap definition, the ‘defineps’ and ‘endps’ keywords tell *pswrap* where a given POSTSCRIPT language program begins and ends. The ‘defineps’ statement defines the resulting procedure call. The *pswrap* translator processes this input and produces a C language source-code file. When compiled and linked with the application, the *PWDisplayText* procedure sends a POSTSCRIPT language program to the interpreter (binary-encoded for more efficient processing), causing the specified text to be displayed.

To summarize, *pswrap* takes a POSTSCRIPT language program as input and gives you back a C language program. After you compile the resulting C program and link it into your application, calling the wrap will transmit a stream of POSTSCRIPT language binary objects to the interpreter.

## 5 THE CLIENT LIBRARY INTERFACE

The Client Library is a linkable library of compiled C procedures that provides an interface between the application and the DISPLAY POSTSCRIPT system. It creates an environment for handling imaging calls to specific Client Library procedures like *DPSmoveto* and custom wraps written for the application.

The Client Library is tailored by the DISPLAY POSTSCRIPT system vendor for your particular operating environment. The *Client Library Reference Manual* describes a generic interface; your system vendor may add additional features or slightly modify the interface to suit your hardware and software. See your vendor’s system-specific documentation for details.

## 6 MANUALS FOR APPLICATION DEVELOPERS

The accompanying set of manuals contains information needed by a POSTSCRIPT programmer developing an application for the DISPLAY POSTSCRIPT system. If you’re new to the POSTSCRIPT language, you should first read the following manuals (published for Adobe by Addison-Wesley and available by calling 1-800-344-8335 or through your technical bookstore):

*PostScript Language Reference Manual*
The standard reference for the PostScript language. Describes the PostScript imaging model and the concepts and facilities of the PostScript interpreter. Documents the PostScript language. Required reading.

*PostScript Language Tutorial and Cookbook*
Provides an informal, interactive introduction to the PostScript language and a collection of programs that offer examples of PostScript usage.

*PostScript Language Program Design*
Gives guidelines for designing and debugging PostScript language programs. Printer-oriented, but some of the information is relevant to writing for a Display PostScript application.

Once you're up to speed in the PostScript language, read the following manuals:

*PostScript Language Extensions for the Display PostScript System*
Describes the extensions to the PostScript language that were made for the Display PostScript system, such as alternative PostScript language encodings, multiple execution contexts, user paths, window system support, and memory management. Introduces important system concepts and documents additional PostScript operators.

*Client Library Reference Manual*
Describes the procedural interface to the Display PostScript system. Tells how to send programs and data to a PostScript execution context, how to handle context output, how to create and terminate a context. Contains procedure definitions, programming tips, and a sample application program.

*pswrap Reference Manual*

Describes how to define C-callable procedures that contain POSTSCRIPT language programs. Tells how to declare input arguments and output to be received from the interpreter. Documents the *pswrap* command line options.

*POSTSCRIPT Language Color Extensions*

Describes color extensions to the POSTSCRIPT language, including multiple color images, color halftone screen definitions, color correction, and CMYK color specification.

Because of the nature of DISPLAY POSTSCRIPT, some of the information you need to write programs is necessarily system-specific. Therefore, in addition to the manuals listed above, you'll need to consult the documentation provided by your system software vendor. The following system-specific documentation is available from Adobe:

*X Window System Programmer's Supplement to the Client Library Reference Manual*

Describes information about the Client Library interface that is specific to the X Window System™, such as context creation and additional error codes.

# Index

Client Library 5

device independence 1
Display PostScript components 1
DPSmoveto 5

imaging model 1, 6

manuals 5

portability 2
PostScript imaging model 1, 6
pswrap translator 3
PWDisplayText 4

window system 1
wraps 2, 3

# How to Order Additional Documentation

## Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

## Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

## Telephone and Direct Mail Orders

| Your Location | Call | Contact |
|---|---|---|
| Continental USA,<br>Alaska, or Hawaii | 800-DIGITAL | Digital Equipment Corporation<br>P.O. Box CS2008<br>Nashua, New Hampshire 03061 |
| Puerto Rico | 809-754-7575 | Local DIGITAL subsidiary |
| Canada | 800-267-6215 | Digital Equipment of Canada<br>Attn: DECdirect Operations KAO2/2<br>P.O. Box 13000<br>100 Herzberg Road<br>Kanata, Ontario, Canada K2K 2A6 |
| International | ——————— | Local DIGITAL subsidiary or<br>approved distributor |
| Internal[1] | ——————— | SDC Order Processing - WMO/E15<br>*or*<br>Software Distribution Center<br>Digital Equipment Corporation<br>Westminster, Massachusetts 01473 |

[1] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).