



The Cypage Operator

Adobe[®] Developers Association

9 October 1997

Technical Note #5608
LanguageLevel 3

Adobe Systems Incorporated

Corporate Headquarters
345 Park Avenue
San Jose, CA 95110-2704
(408) 536-6000

Eastern Regional Office
24 New England
Executive Park
Burlington, MA 01803
(617) 273-2120

Adobe Systems Europe Limited
Adobe House, Mid New Cultins
Edinburgh EH11 4DU
Scotland, United Kingdom
+44-131-453-2211

Adobe Systems Japan
Yebisu Garden Place Tower
4-20-3 Ebisu, Shibuya-ku
Tokyo 150 Japan
+81-3-5423-8100

Copyright © 1997 Adobe Systems Incorporated. All rights reserved.

NOTICE: All information contained herein is the property of Adobe Systems Incorporated.

No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher.

PostScript is a registered trademark of Adobe Systems Incorporated. All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript also is used as a product trademark for Adobe Systems' implementation of the PostScript language interpreter.

Adobe, PostScript, PostScript 3, and the PostScript logo are trademarks of Adobe Systems Incorporated. Apple and Macintosh are trademarks of Apple Computer, Inc. registered in the U.S. and other countries. All other trademarks are the property of their respective owners.

Contents

1	The Copypage Operator	7
	An Overview of the copypage Operator	7
	LanguageLevel 3 Changes to the copypage Operator	7

Adobe Systems Incorporated

Preface

This Document

This is the original release for *The Copypage Operator*, a document that describes how the semantics of the **copypage** operator have been modified for LanguageLevel 3, and explains the likely impact, if any, the modifications will have on applications that currently use the **copypage** operator.

Intended Audience

This document is written for software applications developers who are interested in learning about the modifications to the **copypage** operator, or who may need to make changes to applications that generate PostScript[®] code and use **copypage**.

It is assumed that the developer is already familiar with how the **copypage** operator worked in previous levels of the PostScript language.

Organization of This Document

Section 1, “The Copypage Operator,” covers some background on the **copypage** operator and how it has been modified in LanguageLevel 3.

Related Publications

Supplement: PostScript Language Reference Manual (LanguageLevel 3 Specification and Adobe[®] PostScript 3[™] Version 3010 Product Supplement), available from the Adobe Developers Association, describes the formal extensions to the PostScript language that have occurred since the publication of the PostScript Language Reference Manual, Second Edition. This supplement includes all LanguageLevel 3 extensions available in version 3010.

PostScript Language Reference Manual, Second Edition (Reading, MA: Addison-Wesley, 1991) is the developer’s reference manual for the PostScript language. It describes the syntax and semantics of the language, the imaging model, and the effects of the graphical operators.

Technical Note #5144, "Using EPS Files in PostScript Language Forms," October, 4, 1996, covers a specific use of the **copypage** operator for EPS files contained in forms.

Statement of Liability

THIS PUBLICATION AND THE INFORMATION HEREIN IS FURNISHED AS IS, 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, MAKES NO WARRANTIES OF ANY KIND (EXPRESS, IMPLIED, OR STATUTORY) WITH RESPECT TO THIS PUBLICATION, AND EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSES, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

The Copypage Operator

1 The Copypage Operator

1.1 An Overview of the copypage Operator

The **copypage** operator was originally designed to transmit one copy of the current page to the current PostScript printer without erasing the current page or changing the graphics state in any way.

The **copypage** operator was intended primarily as a debugging aid or as a means of printing successive pages with incrementally accumulated contents. But using **copypage** as a substitute for **showpage** would often degrade the performance of some PostScript printers.

To print multiple copies of the same page (without variable data), it is recommended that the **#copies** implicit parameter of **showpage** or the **NumCopies** page device parameter be used.

To print one or more pages that contain variable data, it is recommended that PostScript language forms be used.

*Note For more information on the **copypage**, **showpage**, **#copies**, or **NumCopies** operators, or for more information on PostScript forms, see the *PostScript Language Reference Manual, Second Edition*.*

1.2 LanguageLevel 3 Changes to the copypage Operator

There are several reasons why the **copypage** operator was changed in LanguageLevel 3:

- Created a great deal of overhead in the design of PS interpreters?
- Inefficient random page access in PS interpreters?
- There is no equivalent in PDF. Pages in PDF have to be independent.

The **copypage** operator now has the same behavior as the **showpage** operator, except that it does not perform an implicit **initgraphics** operation. In particular, the integer argument to the **EndPage** page device procedure indicating the circumstances under which it (**EndPage**) is being called will be 0. This value indicates a **showpage** operation.

Note See Section 4.11.6 of the *PostScript Language Reference Manual, Second Edition*, for more information on the **EndPage** procedure.

The consequences of this change to **copypage** are as follows:

- The code sequence **copypage erasepage** will continue to work as before. Various applications have been observed to do this; the behavior of these applications should not change.
- The code sequence **n {copypage} repeat erasepage**, as a method to produce n copies of the page, will instead produce one copy followed by $(n - 1)$ blank pages.
- The **copypage** operator should no longer be used as a means to implement forms. The first page will print correctly, showing both fixed and variable data. Subsequent pages, however, will show only the variable data, the fixed data having been erased.

Note Because of the changes to the **copypage** operator, certain sections of *Technical Note #5144* no longer apply. Specifically, Section 3, “Emulating 1-page forms in a Level 1 environment,” will mostly be affected.

Because of the modifications to the **copypage** operator, its use is no longer recommended.

Index

Symbols

#copies 7

C

copypage v, vi, 7, 8

E

EndPage 8

erasepage 8

I

initgraphics 8

L

LanguageLevel 3 v, 7

N

NumCopies 7

S

showpage 7, 8