



Don Lancaster

What is PostScript? A Tutorial

The PostScript™ Guest Column

[Editor's Note: this article is an introduction to the PostScript language. In it, Don discusses PostScript features and advantages, fonts, speed and your costs, how to get started and how to buy a new

PostScript printer. Copyright (C) 1990 by Don Lancaster and Synergetics, 3860 West First Street, Thatcher, AZ. 85552. All commercial rights reserved. Personal use permitted so long as this header remains present and intact. Write or call for free PostScript "insider secrets" brochure and product list. Free PostScript helpline 8-5 weekdays, mountain standard time: (928) 428-4073. Or visit <https://www.tinaja.com>.]

What makes Adobe System's PostScript™ language so great? Why has it become the page description language for almost all serious desktop publishing? Maybe we should start off with a simpler question "Just what is the PostScript language?"

Well, PostScript is a new computer language that is normally provided as firmware already built into a PostScript-speaking laser printer or phototypesetter. You most often will not go out and buy a copy of PostScript. Instead, you purchase or lease a ready-to-go PostScript speaking laser printer or phototypesetter. Or else you bolt a new PostScript-speaking lid onto an older and more primitive printer or phototypesetter to upgrade it. Or you can buy PostScript emulators for use on non-PostScript printers.

To use PostScript, all you need is some way of sending ordinary ASCII textfiles to it. These textfiles hold the words and numbers which make up your program. This can be done with any word processor, editor, or comm program on any computer. There are also higher level "power" programs that will auto generate PostScript code for you, plus emulation software that lets PostScript imitate most any older and poorer graphics language.

One key point that most beginners miss is that PostScript is a general purpose computer language. So, most anything you can do in "C", or in BASIC, COBOL, Fortran or Pascal, could also be done with PostScript. Along with any other general purpose computer language, there are some things that do get done quite well and some not so well. PostScript happens to excel at putting nice looking marks onto a printed page.

Thus, besides its being a totally general purpose computer language, PostScript also turns out to be an absolutely outstanding page description language, or PDL, for desktop on up through high-end publishing. PostScript will usually have three different ways of outputting its answer. One is to make marks on a page

and then print that page. The second is to send results back to your host computer via your serial, AppleTalk or SCSI comm channel. And the third is to write files to and from a SCSI hard drive or CD ROM compact disk.

Many beginners often presume that PostScript can only output printed hard copy. And thus they miss two thirds of PostScript's richness.

For instance, I routinely use the PostScript in my LaserWriter NTX with its 12 megabytes of RAM and its 68020 as the "mother's little helper" coprocessor for the 65C02 in my Apple lie. Which really does snap things up a tad. At one time, the controller card in the original LaserWriter was far and away the most powerful computer that Apple built.

What Are The Main Features Of PostScript?

PostScript is related to the Forth language as a third cousin twice removed and seven times disowned. It is an exceptionally easy language to learn, and then quickly becomes totally and permanently addictive. As with Forth, PostScript does use a reverse polish or postfix notation. While weird to look at the first time you see it, this arrangement turns out to let you write exceptionally fast and quite clean high level code.

PostScript is normally interpreted. But it is quite easy to compile or even pseudo compile PostScript sequences for much faster operation.

Two important features of PostScript as a general purpose language is that it is threaded and reentrant. This means that any PostScript command can call any other command, nested to virtually any depth. Thus any command or command sequence in the language can serve as a subroutine to any other command or any command sequence.

A second major feature of PostScript as a general purpose language is that it is extensible. This means you can add as many new commands to the language as you like any way you want at any time. The only catch is that anything new has to get defined through new combinations of already existing commands.

You can also change the name or the meaning of any existing command. For instance, your single new command "book" could automatically print up to 25 copies of your 50 chapter book-on-demand printed math text routed to your customized mailing list, including all equations and all figures and all art, even color photographs if you want to. And do all the invoicing and paper ordering in its spare time.

While there are some 600 commands presently in the PostScript language, you could start doing interesting work by using only a dozen of these. On the other hand, you could easily add thousands of commands of your own, so that the language will become uniquely yours to do with as you please.

A third major feature of PostScript as a general purpose language is that it is stack oriented. A stack is similar to the pile of trays at the start of the cafeteria line. The stack manipulations can be very fast, because