

Why I Like PostScript

Every decade or so, a new idea comes along that just leaps out at you with its freshness, its uniqueness, and its overall vibes. An idea that you just know is inherently "right", absolutely "correct" and above all is and will be "you". Something so elegantly simple that you could really grab on to it. And go with it.

I've personally found the *PostScript* computer language falling into this category of intrinsic wonderfulness. And leading to a mind-boggling array of new opportunities.

If any of your *Midnight Engineering* endeavors have got anything to do in any way, shape, or form with creating some marks upon any sheets of paper, then the *PostScript* computer language cannot be ignored. To do so is sheer folly. I know of nothing that's even remotely close.

PostScript is one universally *general purpose* computer language which can hold its own against any contender. While it happens to excel in dirtying up any otherwise clean sheets of paper, you'll find that page descriptions and graphic layouts form only an inconsequential corner of *PostScript's* real computing abilities.

As one general purpose language, *PostScript* often can end up strongly object-oriented, especially in its *.eps* file transfer modes. There are usually *three* different forms of *PostScript* output, allowing you to make markings upon sheets of thin stuff; to return useful information for host recording, external control, and user interaction; or to write directly to local or shared SCSI disk files.

PostScript is rather highly *device independent*. On the input end, this means that any old text file from any old editor or word processor on any old computer can be used as your program source. On the output end, the very same *PostScript* file can get used to drive anything from a low resolution desktop laser printer up through the fanciest of high precision phototypesetters. *PostScript* can also be used for embroidery, sign cutting, CAD/CAM, video editing, desktop prototyping, plotting, engraving, FAX interchange, image compression, or a host of other new apps.

PostScript is related to *Forth*, as its second cousin three times removed and five times disowned. As such, *PostScript* will use a *postfix* notation (reverse Polish having no goto's); is *extensible* (you can define a virtually infinite variety of new operators as selected combinations of all the existing ones); is *reentrant*, (any operator or proc can call any other one to any reasonable depth); is *stack oriented* (providing for inherent built-in self-addressing); and, very thankfully, is *loosely typed* (you don't need rigid predefinitions).

PostScript is exceptionally good at manipulating strings

and arrays. It is strongly dictionary oriented, giving you local and global constructs. In a mind-boggling feature not available in any other well-known language, its dictionary entries can relate *any two* of dozens of object types.

Not just key-value pairs.

PostScript does its linear graphical transformations on the fly. It is simple for you to translate, to rotate, scale, or otherwise transform any image to any size or any position at any time. *PostScript* has built in cubic spline capabilities which use *Bezier Curves*. These easily let you draw smooth flowing curves given only a sparse set of the ending and control points. Your use of these splines lets you create typography with use of a procedural description rather than requiring a large bitmap. Only your single outline description (combined with suitable hinting rules) gets needed for all sizes and all variations of any user-chosen font. Unlike the *hershey bar effect* of scaled bitmaps, scaled splines will get *smoother* as they are enlarged.

Sophisticated font machinery is an inherent feature of *PostScript*. A two-step process is used. The very first time a character is needed, it will get built up from its tightly compacted description. At the same time, a bitmap copy of the final character is usually saved to a *font cache*. Repeated reuse of the same character is much faster.

The latest *PostScript Level II* adds some surprising new features, such as a full FAX capability, instant and totally unlocked access to all of your font paths, higher speeds, improved color seps, stronger forms and step-and-repeat capabilities, JPEG video compression, and more.

As a general purpose computer language, *PostScript* is fast and fun to apply. You can generate award winning output with nothing but five minutes of instruction.

Or you could spend your whole lifetime exploring and pushing the envelope of this incredibly compelling new multimedia communications tool.

Be forewarned: *PostScript is extremely addictive*.

You do not normally go out and buy your very own copy of the *PostScript* language. Instead, the language is built into or gets added on to most popular laser printers. It is important that you realize that your *PostScript* speaking printer is really a general purpose computer which executes *PostScript* programs. But an extremely powerful computer having a fast and an exceptionally clean architecture that is lots of fun to program and use. This is definitely *not* a "just dump the text" printer. And does not become one until such time as you or some firmware teaches it exactly what you want it to become.

Programming in PostScript

A PostScript program is typically nothing more than a plain old ASCII textfile which contains ordinary words and numbers. A PostScript program could arrive *preinstalled* as firmware or in cartridges; can be *downloaded* on the fly; or can be *persistently downloaded* once to remain available as long as the computer remains powered. Downloading can be done with a serial port, AppleTalk, shared SCSI comm, Ethernet, modem, hard disk, or CD ROM.

But do note that ordinary *one way* parallel ports must *not* be used with PostScript, since they will blindfold you and tie both of your feet together. Proper use of PostScript *demand*s an instant two-way comm channel.

There are several popular ways of generating PostScript code. One is to use *raw PostScript* to create your very own programs by using your favorite word processor or editor. This route is my overwhelming favorite and is by far your most powerful and your most flexible. Raw PostScript also usually generates short and fast-running files.

A second method for obtaining PostScript code is to use an *emulator*. This can be any currently running PostScript program which accepts an input in some selected format and then will convert it on the fly into suitable PostScript commands. The popular traditional emulators do include PCL, *Diablo*, *Imagewriter*, *Epson*, HPGL, and TEX. Most users usually outgrow traditional emulators twelve minutes or less into their use. ("Gee. I'd sure like a few larger drop titles, an initial cap, proportional spreadsheet fonts, and a box around that sidebar...")

PostScript Level II adds powerful new *filter* emulators. Early filters include *fax* and *JPEG* video.

Your third method for generating PostScript program code is to use an *applications package* to write the code for you. The obvious examples here include *Pagemaker*, *Adobe Illustrator*, *Ventura Publisher*, and *Aldus Freehand*.

Although fast and convenient for beginners, a lot of these application packages force you to do their own things in their own way and generate code that can be inherently slower, lots longer, and far less flexible than by writing your own raw PostScript.

EPS files are related to the PostScript program generators. These can be pretested and (hopefully) debugged textfile program modules that have a standard header and trailer. These also disallow some obscure PostScript operators that can cause havoc when imported into another program. .EPS stands for *Encapsulated PostScript*.

Send in the Clones

To date, there have been bunches of attempts at faking PostScript, done by everyone from several free shareware groups up to Apple's and MicroSoft's abortive *TrueType* ludicrousity. The latter is finally and mercifully being left to quietly rot in their collective back yards.

I've done extensive testing of most versions of PostScript and most of these fakes, and have *never* found a fake that could hold a candle to the real thing. *All* of them, without exception have got serious compatibility and performance problems. To date, there is *not one* of them that I have not been able to prove incompatible in twelve minutes flat.

The only thing which prevents a conviction on claiming "100 percent PostScript compatibility" is that the jury ends

up on the floor in hysterics. In fact, a good definition of "PostScript Compatible" is that the prongs on their power cord are roughly the same size and shape as those on a real PostScript printer.

Several very useful things these fakes have done include forcing Adobe to reduce prices, to improve and extend the documentation, and to unlock all their font paths.

The fakes can, in theory, get you started with something vaguely like the way PostScript used to be at low costs. And do so on older and more traditional printers.

But most end users very quickly outgrow a fake.

At any rate, there is a freeware fake known as *GhostScript* downloadable by way of www.tinaja.com/post01.html and elsewhere. Included are the run time modules, "C" source code and several fonts.

Genuine PostScript is available through *Adobe Systems* at www.adobe.com Adobe has an active developer program.

UPDATE: *GhostScript* has gotten a lot better. And *Adobe Acrobat .PDF* runs PostScript on most any printer. *Distiller* is a superb host-based PostScript interpreter.

Emerging PS Opportunities

A general purpose computer language could do anything for anybody, especially when it has an exceptionally strong graphics ability. Thus, you could apply PostScript to most anything you care to. Here are a few of the directions that PostScript seems to be taking me at present...

Book-on-demand Publishing— The printing and binding of books only when and as ordered can revolutionize the entire publishing industry, especially when combined with CD ROM. The economics here are absolutely outstanding, especially for low quantities or unknown total sales.

EPS BBS Downloads— Hobby tech magazines have faced a big dilemma for years: How can first rate printed circuit artwork, dialplates, and layout guides be placed directly in the hands of their end users? PostScript modemed EPS files are an obvious answer. An answer that can work with any word processor on any computer. And by first going to a *Distiller* pseudocompile, no trace whatsoever remains of any proprietary source code.

Universal PS frame devices— A framedevice is simply the bitmap any PostScript interpreter produces. By making this bitmap readily available, PostScript can get extended to zillions of those new application areas. Such as animation stands, printed circuit drills, routers, sign makers, Santa Claus machines, NC milling, automated looms, CAD/CAM, embroidery stitchers, woodworking lathes, etc...

Engineering graphics— PostScript seems especially adept at fitting very smooth curves to fuzzy or sparse data sets. This makes your data sheets for semiconductor responses or pump characteristics simple to do.

Super FAX— The level II PostScript now can include a full FAX formatting capability. By combining this with some external smarts, the PS printer can either send "perfect" fax to your remote fax machine, or else send a much higher quality and more compact device independent PostScript to a similar machine at the other end. Thus, you can now send camera-ready graphics and other top-notch artwork faster, cleaner, and cheaper than grubby old FAX.

Printed circuit design– PostScript is ideal for most printed circuit layouts, especially when integrated with some direct toner method process that uses the toner itself for a final etch resist. PostScript can also emulate existing Gerber and HPGL photoplotter files.

Stock analysis– PostScript can be a totally general purpose language. As such, it is well suited to such financial uses as a stock market performance analysis. Especially valuable is the ability to take tabular values extracted from any source and convert into attractive and meaningful graphs. Either on a manual or a fully automated basis.

Video editing– Others are now working into the obvious extensions of PostScript into 3-D animated graphics, video production, and editable modules suitable for multimedia uses. What PostScript does to printed images, it should ultimately be able to do to sight and sound. It makes sense to have one single language that handles screen images, printed hard copy, and full motion video.

Speed issues– A dollar can be made in exploring some obvious PostScript speedup tricks. Because most users (and virtually all applications) go far out of their way to baud rate limit themselves when they first start to use PostScript. PostScript can end up ridiculously faster than most users suspect. The opportunities here now include understanding and exploring the PS *Click-to-clunk* curve, going to *Ethernet* or *Shared SCSI Comm*, extending the hard disk usage, and further improving program pseudo compiling.

Refills and rebuilds– There are now millions of PostScript printers out there. Possibilities abound in used machines, upgrades, rebuilds, service, and training. Knowledgeable users can do all of their own better-than-original cartridge refills for \$5.50, and reducing their per-page toner costs by 25:1 or more. Many support products remain outrageously overpriced or else monumentally mismarketed.

Open font paths– Because the earlier versions of PostScript stupidly locked out font paths, you had to go to extreme lengths, ridiculous sneakiness (or both) to extract your font paths for such non-linear transformations as star wars or perspective lettering, text on an arbitrary surface, twisted film effects, and such. The latest of level II has completely unlocked font paths which should tremendously simplify such tasks. And open a whole new world.

Step & repeat and forms– The PostScript language always has had a strong forms and step-and-repeat capability. In the past, this has been especially handy for such things as shipping labels, where you leave the background art intact and erase only the old name and address each time. Level II now adds explicit new forms capabilities that should be especially useful in duplex printing applications.

Custom Logos and such– There is a nearly limitless local market for custom smaller quantity PostScript design and layout work. Stuff a traditional printshop wouldn't touch. Such as five sequentially numbered bumper stickers. Or a dozen "here's how you can get to our house" business cards having maps on them. Or ten wedding invitations. Or a custom rubber stamp. Or fundraiser badges. Or T-Shirts. Or gimme caps. It's all a question of developing your own local market. And then creatively mining it.

For More Help

Two essential PostScript books are the Adobe *blue* book, and otherwise called the *PostScript Tutorial and Cookbook*, and the revised *red* book, titled as the *PostScript Reference Manual II*. That red book includes full details on EPS files, Level II, and on the new *Display PostScript*.

LaserWriter tech info appears in the Apple *white* book, also known as the *LaserWriter Technical Reference*.

I am also now laboring under the delusion that my *Intro to PostScript* video and my *LaserWriter Secrets* book/disk combo might be of help to you. I do carry bunches of other PostScript products from Adobe, by myself, and by other leading authors. There's now a no-charge PostScript voice helpline and forum here at my *Synergetics* which you can call per the trailer below.

Let's hear from you. ♦

UPDATE For the latest developments on PostScript and its newer Acrobat variant see www.tinaja.com/post01.html and my www.tinaja.com/acrob01.html

Microcomputer pioneer and guru Don Lancaster is the author of 35 books and countless articles. Don maintains a US technical helpline you'll find at (520) 428-4073, besides offering all his own books, reprints and consulting services.

Don has a free new catalog crammed full of his latest insider secrets waiting for you. Your best calling times are 8-5 weekdays, Mountain Standard Time.

Don is also the webmaster of www.tinaja.com You can also reach Don at Synergetics, Box 809, Thatcher, AZ 85552. Or you can use email via don@tinaja.com

PLEASE CLICK HERE TO...

-  **Get a Synergetics catalog**
-  **Start your tech venture**
-  **Sponsor a display banner**
-  **Find research solutions**
-  **Send Don Lancaster email**
-  **Pick up surplus bargains**
-  **Find out what a tinaja is**
-  **View recommended books**