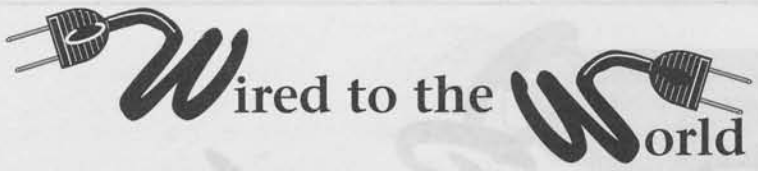

by *Ralph Lee Scott*

Java and the Web

The *Java* language came on the software market in 1995. *Java* is a programming language like C++ or Fortran that creates or compiles instructions that tell your computer what to do. It has a number of features that make it an improvement over other compilers. The chief of these features is that *Java* programs work by downloading code (or instructions) from the Internet in small doses called *Applets*. The main advantages of this applet system are: 1) these small programs, after initial download from the Web, run on your computer without going back to the Web page for updates; 2) these programs run faster because they do not have to go back to the Web host computer (which might be busy or off line) to post information; 3) the *Java* programs run in real time, thus you can download and run software on your machine as well as just viewing Web pages; 4) the applets use a GUI (Graphical User Interface) so things like buttons, mouse movement, text boxes, etc. can be referenced.

Among other useful aspects of the language as expressed in a *White Paper* written by the developers of *Java*, the Sun Microsystems Corporation, is that it is "simple, object oriented, distributed, secure, robust, portable, platform neutral, interpreted, high performance, multi-threaded and dynamic." In short, it works, we hope, safely when we need it. A lot of software is currently under development using these *Java* technologies. For example: Microsoft's *Exchange* e-mail software uses *Java* applets to send and receive messages through an Internet proxy server. This enables the system to remain secure, but also allows e-mail users around the world to access their mail system. Another example of the use of applets is in MSNBC (Microsoft NBC) home page Web site (*www.msnbc.com*). While the reader is looking at the MSNBC home page on his browser, news applets are running the background giving updates to current headline, weather, and folders of interest. Another Web application of applets is the KPIX traffic page (*www.kpix.com/traffic*), where real time TV cameras broadcast current San Francisco freeway traffic conditions worldwide. To view these *Java* applications, you must have what is called *Java* enabled Web browsers. The two current major Internet browsers, Netscape's *Navigator 4.0* and Microsoft's *IE 4.0*, both support *Java* technologies. The only catch is that you must have the *Java* script turned on through a pull-down software switch.

Major detractors of the *Java* technology argue that the language is a Sun proprietary product and not a standardized compiler. This, of course, allows one company to hold a monopoly on product development. Developers of standardized languages like C++ and Fortran argue that by developing language platforms cooperatively over many vendors, one gets a better product — a product built on mutually agreed goals. This way the end user is not forced to use just what the proprietary developer thinks is best. Some programmers with this bent have even been known to state that "*Java*? You don't need no steenken' *Java*."[1] Other programmers/users contend that proprietary software has the advantage of more focused, product-oriented goals. In other words, the software works better because the company has a vested interest in creating and selling a good-working product.

Sun Microsystems' *Java* has a powerful ally, Netscape Communications, which has recently introduced what it calls the "Visual Basic of the Web." *Basic* was a simple programming language that most computer users studied as their "first language." Netscape's new *Java* development software, which was just released in November at COMDEX, is called *Visual JavaScript*. *Visual JavaScript* used click and drag icons to create components that run as *Java* applets, *Java*-Beans, HTML code, or COBRA (Common Object Broker Architecture). These components are combined by *Visual JavaScript* into what Netscape calls *JavaScript Beans*. The "Beans" are joined by the software into the finished Web page, using an application called Connection Builder.[2] The *Visual JavaScript* software is designed to work with Netscape's Web development tool package, *SuiteSpot*.

Java has its limitations, the major one being that it slows down the loading of the page on your desktop. Another drawback is that the *Java* applets cannot access data stored on a server, or modify an entire Web page once it has been sent. Our good friends at Microsoft tried to solve the one-way nature of *Java* by the use of control features in the IE browser called *ActiveX*. *ActiveX* actually takes control over a section of the desktop and can display information and respond to commands from within the browser display without actually reloading the entire page. There are, of course, limitations to what *ActiveX* can do, primarily due to a lack of platform portability (i.e., it does not come with Netscape Navigator), and size/security concerns with your computer desktop. The latest transport software like *Java* under development is called *Dynamic HyperText Markup Language* (or *DHTML*). *DHTML*, like *Java* and *ActiveX*, promises to change the way the Internet works. With *DHTML*, once a page has been loaded on your desktop, you can interact with software to send with the original page without going back to the server. (Remember this was the original idea of *Java*.) With the new *DHTML* technology, additional pages, or parts of pages, remain hidden to be called up locally when you need them. The best analogy I can think of for *DHTML* is that it is like a big flip chart. On your desktop you can flip back and forth among pages, parts of pages, graphics, text, and even varying levels of content. More about *DHTML* in a future "Wired" column.

References

[1] Paul Kapustka, "Geek Patrol: *Java* Gets a Scalding," *Internet Week* (November 10, 1997): 16.

[2] Ellis Booker, "*JavaScript* Becomes Visual," *Internet Week* (November 17, 1997): 12.