Implementing Advanced JavaScript Techniques

We're assuming the audience of this book has at least a basic working knowledge of JavaScript. A complete tutorial on JavaScript fills a complete book itself, so we'll avoid trying to teach the language here. Instead, this section discusses some of the advanced and possibly little-known features of JavaScript and how you can incorporate them into your Ajax development.

We'll first cover a little history about JavaScript so you know where it has been and how it got here. Brendan Eich of Netscape developed JavaScript in 1995. His task was to develop a way to make Java applets more accessible to the nontechnical Web designers who created and maintained Web sites. Eich decided that a loosely typed language devoid of compilers was the appropriate choice.

Various names were attached originally to Eich's creation, but it was finally renamed to JavaScript in an effort to capitalize on Java's newfound marketing success. JavaScript swiftly became the most popular scripting language on the Web, thanks to a low barrier of entry and an ability to be copied and pasted from one page into another. Early revisions of JavaScript and the Navigator DOM gave rise to the DOM Level 0 standard, which defined form elements and images as children of their elements.

Not to be outdone, Microsoft created its own scripting language called VBScript. VBScript was functionally similar to JavaScript but had a Visual Basic–like syntax and worked only in Internet Explorer. Microsoft also supported an implementation of JavaScript (which by now had been and standardized and named ECMAScript by ECMA) as JScript. While the syntax of the various flavors of JavaScript were nearly identical, the vast differences in the implementations of the DOM among browsers made cross-browser scripts almost impossible to create. Using a lowest common denominator approach usually led to scripts that could do no more than the most trivial of tasks.

By 1998 Netscape had opened the source code for its browser and decided to rewrite the browser from scratch with a focus on closely following W3C standards. At that same time, version 5 of Internet Explorer had by far the best implementation of the W3C DOM and ECMAScript. The first complete release of the open-source Netscape code under the Mozilla banner came in 2002. This started a trend in the browser space: more and more browsers worked to comply with Web standards maintained by the W3C and ECMA. Today, modern browsers such as Firefox, Mozilla, Opera, Konqueror, and Safari all adhere closely to Web standards, greatly simplifying the task of writing cross-browser HTML and JavaScript. Internet Explorer 6, not much changed from the version 5 browser of 1998, exhibits the most nonstandard behavior.

Object-Oriented JavaScript via the prototype Property

JavaScript supports a form of inheritance through a linking mechanism rather than through the classical inheritance model supported by fully object-oriented languages such as Java. Each JavaScript object has a built-in property named prototype. The prototype property holds a reference to another JavaScript object that acts as the current object's parent.

An object's prototype property is used only whenever a function or property of the object is referenced via the dot notation method but is not found on the object. When this situation occurs, the object's prototype object is inspected for the requested property or function. If it doesn't exist on the object's prototype property, then the prototype's prototype is examined and so forth up the chain until the requested function or property is found or the end of the