CHAPTER 9

Widget Mania: Using a GUI Widget Framework

In web development, widgets are all the rage these days. No longer are we content with regular form fields, buttons, drop-down lists, and such. Just plain-old tables aren't sufficient for many developers! But this trend isn't just due to a desire to create cooler interfaces. There are UI metaphors in a modern operating system that don't have an analogy in the web world, at least not intrinsically.

Take the tree view as an example. I'm sure you've seen a tree view before. Indeed, if you work in Windows, you can barely avoid it (it's the list of folders on the left in Windows Explorer). Have you seen one on the Web? Chances are you have, but unless it was relatively recently, the web developer likely wrote it himself, or else lifted some code from somewhere else. What it wasn't though, in all likelihood, was a piece of code that was self-contained and part of a larger framework of UI components, which is what a UI widget is.

Widgets make adding this "advanced" functionality to your web application very easy, and more important, consistent. You'll see how this works as we build a handy little application for posting notes to ourselves. In this project, we'll use a lot of widgets supplied by a library that is rapidly growing in popularity: the Yahoo! User Interface (YUI) Library.

JSNotes Requirements and Goals

For this chapter's project, which we'll call JSNotes from here on out, we're going to build something along the lines of that pad of sticky yellow notes you very possibly having sitting around your desk somewhere. In this case though, it will be a web-based application that allows us to post digital notes to ourselves. The main purpose of this application, aside from being useful in helping to keep us from losing little pieces of information we obtain throughout the day, is meant to demonstrate the use of the YUI Library.

So, what specifically is JSNotes going to do? Let's spell it out now:

- JSNotes will allow us to create notes, including a subject for each, and add a date and time (generally, the current date and time, but you never know!). We'll also be able to categorize each note as either personal or business.
- The notes will be presented in a Windows Explorer-like view, with a tree view on the left and note details on the right. The two primary branches in the tree will be our two categories: Personal and Business.

- For each note, we'll store the note text, a subject, and a date and time, as well as the note's category.
- We'll be able to delete a note, and we'll also be able to "export" a note, which really just means put it in a form suitable for copying and pasting into another program.
- When adding a note, we want to present it in a pop-up dialog box.
- We'll also have Help and About boxes, but they will be done in a different style than the add note pop-up. These will appear as an overlay over the main JSNotes display.

The best part is, with the YUI Library in the mix, all of this becomes a relatively trivial exercise!

The YUI Library

The YUI Library includes a set of UI widgets, as well as a number of utility-type classes for Ajax, drag-and-drop, DOM manipulation, and more. YUI is one of the best documented libraries I've seen recently and also one of the best demonstrated. You can find a good example of virtually every part of the library, usually along with a number of variations for guidance.

Let's talk about the widgets, since they are primarily what we'll be working with in this application, and we'll be using quite a few of them. YUI provides a number of widgets, including AutoComplete, Calendar, Container (including Module, Overlay, Panel, Tooltip, Dialog, and SimpleDialog), Logger, Menu, Slider, TabView, and TreeView. Because the widgets are designed around a common framework, they all present a fairly consistent programming interface.

Using YUI is as simple as importing some JavaScript files, and in many cases, some CSS files as well. Generally speaking, each widget is a single JavaScript file, but there also may be some dependencies on other files, and those need to be manually imported, too. Once that's done, you're ready to rock and roll.

For instance, let's say you want to create a menu. All you need to do is define your menu in the form of some simple markup, like so:

```
<div id="basicmenu" class="yuimenu">
  <div class="bd">
    <ul class="first-of-type">
      <li class="yuimenuitem"><a href="page1.htm">Page1</a></li>
      <li class="yuimenuitem"><a href="page2.htm">Page2</a></li>
    </ul>
  </div>
</div>
```

This provides the basic structure of the menu. The last step is to tell YUI to create the menu for you, based on this markup. To do so is as simple as this:

```
var oMenu = new YAHOO.widget.Menu("basicmenu");
oMenu.render();
```