

Using popupmenu and AeB Pro  
to create Popup Menus  
with Layers

D. P. Story

The  $\LaTeX$  source file for this [AeB Blog](#) is attached to this document. [Click here](#) to open the attachments panel. The file uses AeB Pro, so Adobe Distiller, 7.0 or later is required to make the PDF. If you don't have Distiller, there are some interesting JavaScript code to study.

The latest version of the popupmenu package is an attachment to the blog article [Creating Popup Menus with the popupmenu Package](#).

## Layers, Links, and Popups

Rollover the button below to learn about some of the fine quality products available to your free-of-charge from the AcroTeX Family of L<sup>A</sup>T<sub>E</sub>X packages.

---

The AeB Family of fine L<sup>A</sup>T<sub>E</sub>X software has been under development for more than a decade, and there a numerous members in the family. The main thrust of the work is to provide educators tools for creating high-qualify, interactive materials in the Portable Document Format (PDF).

With AeB and AeB Pro, you have have the tools to create a design for your document, to develop a document that the user interacts with, and to control the PDF viewer of your interactive PDF file.

You can use AeB to create online exercises and quizzes. The quizzes are graded using embedded JavaScript, and solutions can be optionally provided. AeB Pro provides access to more advanced features, such as layers, attachments, and multimedia (Flash SWF and FLV files), all through a L<sup>A</sup>T<sub>E</sub>X authoring system.

With AeB and AeB Pro, you are only limited by your own imagination.

## Accessing the title with a return value

When there is no return value specified, the title of the menu item is returned. When there is a return value and the title is not incorporated in the return value, how can you get the title of the menu anyway? Here's how.

We use the same AeBMenu menu structure example as was used earlier. The return value of this menu structure does not contain the name of the menu item, yet, the title of the menu item appears. The code for the mouse over action is

```
var aChoice=processMenu(AeBMenu);
if (aChoice!=null) {
    var thisChoice=aChoice[0];
    var thistitle=eval("AeBMenu"+thisChoice+".cName");
    app.alert(thistitle);
}
```

The first entry in the return value is the offset into the AeBMenu array of the user's choice. We save this as `thisChoice`. The entries of the return value are strings so to access the title, we execute

```
var thistitle=eval("AeBMenu"+thisChoice+".cName");
```

When the string is evaluated with the `eval` method, it becomes the correct reference to the menu/submenu item the user selected. We tack on `.cName` to retrieve the value of the `cName` property (which corresponds to the value of the `title` key in  $\LaTeX$  of the selected menu item).

Now, back to my retirement!