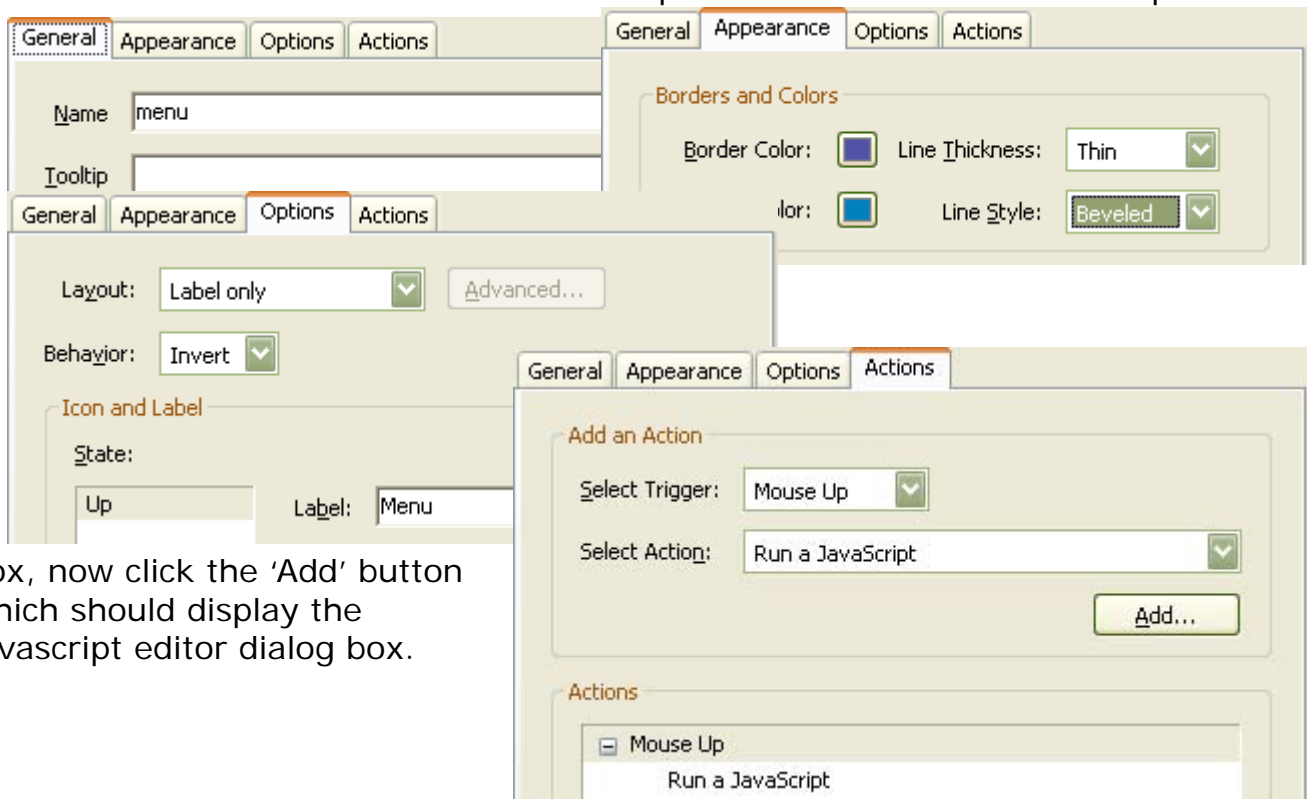


The Popup Menu has been around since Acrobat version 4 and has often been overlooked as a means to enable your users to get around your documents quickly or to activate Acrobat menu items or features .

To create a Popup Menu let's start by creating a form field button. Name the button something like 'menu' or 'popup'. Format the button so it has a border and a fill colour, set the Line Style to be 'Beveled'. In the 'Options' tab enter a meaningful label such as 'Menu' or 'Navigation'.

Figure 1, Figure 2, Figure 3, Figure 4 & Figure 5
Under the Action tab select 'Run A Javascript' from the 'Select Action' dropdown



box, now click the 'Add' button which should display the javascript editor dialog box.

```

Create and Edit JavaScripts
var cChoice = app.popUpMenuEx
(
    {cName: "Item1", bMarked:true, bEnabled:false},
    {cName: "-"},
    {cName: "Item2", oSubMenu:
        [ {cName: "Item2 Sub1"},
          { cName: "Item2 Sub2", oSubMenu:
              {cName:"Item 2 Sub2 Subsub1", cReturn: "0"}
            }
        ]
    },
    {cName: "Item3"},
    {cName: "Item4", bMarked:true, cReturn: "1"}
);
    
```

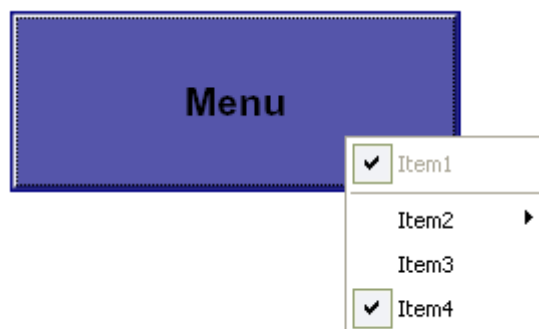
The basic structure of the popup menu is as follows:

1. Use the 'app' object to call the popup menu function: 'popUpMenu'
2. Pass parameters to the function indicating:
 - a. Menu Item Name, what get's displayed (cName)
 - b. Whether or not the Menu item is a checkbox (bMarked: true/false)
 - c. Is the menu item enabled or disabled? (bEnabled: true/false)
- d. A Sub Menu, which is created in the same fashion as a normal menu, this parameter is optional and only required if you want sub menus. (oSubMenu: {...})

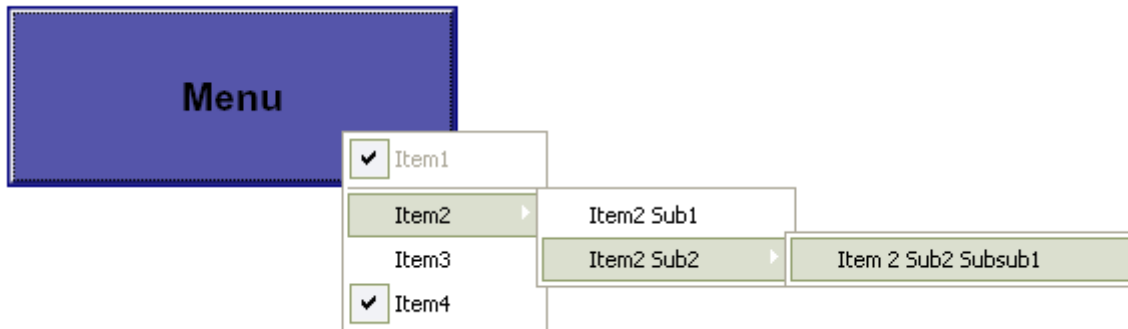
If you paste the following code into the Javascript Editor, you'll get the following Popup Menu

```
var cChoice = app.popUpMenuEx
(
  {cName: "Item1", bMarked:true, bEnabled:false},
  {cName: "-"},
  {cName: "Item2", oSubMenu:
    [ {cName: "Item2 Sub1"},
      { cName: "Item2 Sub2", oSubMenu:
        {cName:"Item 2 Sub2 Subsub1", cReturn: "0"}
      }
    ]
  },
  {cName: "Item3"},
  {cName: "Item4", bMarked:true, cReturn: "1"}
);
```

Figure 6 & 7



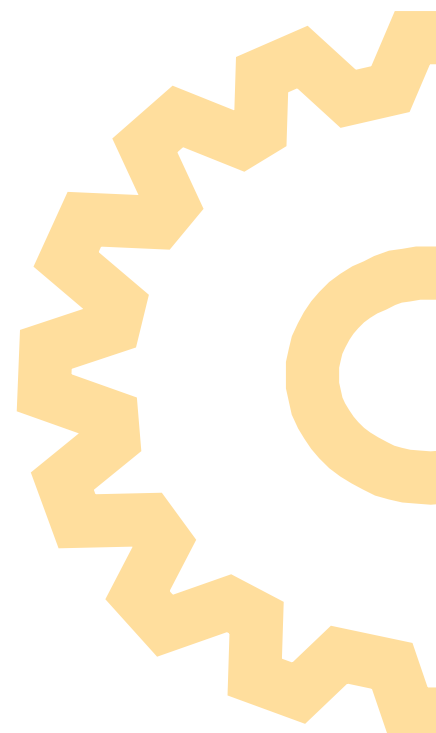
As you can see the Menu Items in the popup menu have a direct relation to the example code we pasted above.



Something more meaningful could be to have the popup menu go to key pages in your document, for example you may have a Table of Contents page along with an Index or Glossary page. You could construct a Menu that showed the names of each of these sections and returned either the page number or named destination they appeared on.

For example, let's assume we have created named destinations for each of the sections in our document:

Table of Contents:	"toc"
Index	"index"
Glossary	"glossary"
Help	"help"



We also want to have a quick print function

```
var cChoice = app.popUpMenuEx
(
  {cName: "Table of Contents",cReturn: "toc"},
  {cName: "-"},
  {cName: "Index",cReturn: "index"},
  {cName: "Glossary of Entries",cReturn: "glossary"},
  {cName: "Help Pages",cReturn: "help"},
  {cName: "Quick Print",cReturn: "qp"}
);

if(cChoice != "qp" && cChoice != null)
{
  this.gotoNamedDest(cChoice);
}
else if(cChoice != null)
{
  this.print({bUI:false,bSilent:true});
}
```

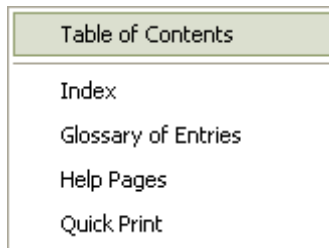


Figure 7