

# Top Ten JavaScript Enhancements for Help

In this paper, I describe ten JavaScript scripts that I have created for various help projects. For each script, I have included a browser compatibility chart, a short description, any caveats or "gotchas" involved, and a code sample.

All of these scripts are available on the User First Services website at [www.userfirst.net/demos](http://www.userfirst.net/demos). I have included a copyright statement in the more complicated scripts. If you would like to use them, please contact me at [scott@userfirst.net](mailto:scott@userfirst.net) or 770.446.8474. I would like to track how they are being used, and I would like to keep a contact list to notify people of any changes.

## A few JavaScript tips

If you type in the code samples, make sure you type everything **exactly** as it appears. JavaScript is as strict as my ninth grade English teacher--one punctuation error and you fail. Braces, parentheses, and brackets are easy to confuse with each other. Capitalization and spelling definitely count. I'd recommend downloading the scripts from my site. It'll save some time and frustration. Also, Mrs. Stowers (my teacher) didn't review this document. I tested everything twice, but there's still a chance I made a typo.

## A note about browser compatibility

I have tested each script in Windows 98 using Internet Explorer 4 and 5, Netscape 4 and 6 (Preview Release 2), and Opera 5. Where applicable, I have also tested the scripts with HTML Help v1.2 in Windows 98. The Preview Release 2 version of Netscape 6 is not very stable, so some of the scripts marked as not compatible with Netscape 6 will probably work in the final release.

## symbols and their meanings

- = works
- ◉ = works, but not exactly as designed
- = does not work
  
- ↵ = same script line (do not press Enter)

OK, let's start scripting!

## Opening a help window with the F1 key

---

- IE 4
- IE 5
  
- Netscape 4
- Netscape 6
  
- Opera 5

This script opens your help system (rather than the browser's help system) when the user presses F1. It uses the onHelp event in the BODY tag to open the help. The "return false" statement tells the browser to not open its own help system.

**Note:** The onHelp event is only supported in Internet Explorer. Netscape and Opera will only open their Help systems when you press F1.

```
<html>
<head>
<title>Opening help with the F1 key</title>

<script>function openHelp() {
window.open('helppage.htm', ↵
'helpwin', 'toolbar=0,location=0,directories=0,status=0, ↵
menubar=0,scrollbars=0,resizable=0,width=520,height=350');
}
</script>
</head>

<body onHelp="openHelp();return false">
Press F1 to open the help.
</body>
</html>
```



## Opening context-sensitive help topics

---

- IE 4
- IE 5
- Netscape 4
- Netscape 6
- Opera 5

This script opens context-sensitive help topics for Web-based applications without using map files. Instead, it expects the help page's filename to be based on the application's name. For example, the demo.htm page would have a help page called "h\_demo.htm."

```
<html>
<head>
<title>Opening context-sensitive help topics</title>

<script language="JavaScript">
function openHelp() {
helpurl = location.href;
begin=helpurl.lastIndexOf('/');
begin = begin + 1;
end=helpurl.lastIndexOf('m');
end=end + 1;
helpurl = "h_" + helpurl.substring(begin, end);
window.open(helpurl, 'helpwin', 'toolbar=0,location=0, ↵
directories=0,status=0,menubar=0,scrollbars=0,resizable=0, ↵
width=520,height=350');
}
</script>

</head>

<body>
<a href="javascript:openHelp()">open help window</a>

</body>
</html>
```

## Showing mouseover help

---

- IE 4
- IE 5
- Netscape 4
- Netscape 6
- Opera 5

This script uses the `onMouseOver` and `onMouseOut` events to show and hide field-level help topics. The links send the help layer's id to the script (in this example, it's called "fieldhelp").

```
<html>
<head>
<title>Showing mouseOver help</title>

<script>
function showhelp(helpid) {
if (document.all) {
document.all(helpid).style.visibility = "visible"
}
else { document[helpid].visibility = "show" }
}

function hidehelp(helpid) {
if (document.all) {
document.all(helpid).style.visibility = "hidden"
}
else {
document[helpid].visibility = "hide"
}
}
</script>

</head>

<body>
<a href="#" onmouseover="showhelp('fieldhelp') "
onmouseout="hidehelp('fieldhelp') ">mouse
over this field</a>
<div id="fieldhelp" style="position:absolute; width:200px;
height:115px; z-index:1; visibility: hidden; background-color:
#FFFF99">some Help </div>

</body>
</html>
```

## Using autotext fields

---

- IE 4
- IE 5
- Netscape 4
- Netscape 6
- Opera 5
- HTML Help

This script allows you to store commonly used information as "autotext" entries.

To create autotext entries, create an external JavaScript "js" file. In the js file, add:

```
function autotext(entry) {
if (entry == 'author') document.write('Scott DeLoach');
}
```

In your topic, use the following code to reference your autotext entry:

```
<SCRIPT>autotext('author')</SCRIPT>
```

where `author` is the name of the autotext entry.

## Automatically creating breadcrumbs

---

- IE 4
- IE 5
- Netscape 4
- Netscape 6
- Opera 5
- HTML Help

Breadcrumbs are used to show the user where they are in a help system (for example: Top Ten JavaScripts : Breadcrumbs). This script adds automated breadcrumbs to your help topics. The script simply uses your folder structure to build the breadcrumbs. I am using a replace script to change underscores to spaces and slashes to colons.

```
<html>
<head>
<title>Automatically creating breadcrumbs</title>

<script>
function replace(string,text,by) {
var strLength = string.length, txtLength = text.length;
if ((strLength == 0) || (txtLength == 0)) return string;
var i = string.indexOf(text);
if ((!i) && (text != string.substring(0,txtLength))) return
string;
if (i == -1) return string;
var newstr = string.substring(0,i) + by;
if (i+txtLength < strLength)
newstr +=
replace(string.substring(i+txtLength, strLength), text, by);
return newstr;
}

// breadcrumbs
// (c) 2001 Scott DeLoach, User First Services
hansel = 1;
trail = location.pathname;
begin=trail.lastIndexOf(':');
begin=begin+2;
end=trail.lastIndexOf('/');
trail = trail.substring(begin, end);
if (trail.length == 1) hansel = 0;

newstring = replace(trail,'/',' : ');
newstring = replace(trail,'_',' ');
newstring = replace(trail,'%20',' ');
</script>

</head>

<body>
<h1>Sample breadcrumbs</h1>
<script>if (hansel != 0) document.write(newstring + "<BR>");
</script>

</body>
</html>
```

## Supporting client annotations

---

- IE 4
- IE 5
  
- Netscape 4
- Netscape 6
  
- Opera 5
  
- HTML Help

This script allows site owners to add annotations to help topics. It does not allow users to add annotations, but it could be extended to support user annotations. I have been using this script to provide help for customizable web applications.

### How it works

When you open a help page, say "help.htm," the script tries to open help\_custom.htm in a hidden frame. After the hidden frame loads, it triggers a script that checks to see which page is loaded in the hidden frame. If help\_custom.htm actually loaded, the layer that contains the annotations paperclip is made visible. If the help\_custom.htm file did not load (which means a 404 message file loaded), then the paperclip remains invisible.

The script expects your help topics to appear in a frameset with a top frame for the content and a bottom frame for the hidden annotation topics. In my sample code, the top frame is called "ftop" and the bottom frame is called "fbottom." The initial bottom frame topic uses JavaScript to load a one frame frameset:

```
<html>
<head>
<title>notes</title>
</head>

<SCRIPT LANGUAGE="JavaScript1.1">
url = "blank.htm";
if (location.search.substr(1).length > 0) url =
(location.search.substr(1));
document.write("<frameset rows='100%,*' name='noteset' cols='*'
onLoad='top.ftop.loaded()'>");
document.write("<frame src= '" + url + "' name='note'>");
document.write("</frameset>");
</SCRIPT>

<BODY>
</BODY>
</HTML>
```

When a topic opens, it sends the name of the matching annotation file as a search string with the url to the bottom window (that is done by loading a file as help.htm?searchinfo). The search string is often used by search engines and e-commerce sites to pass data between pages. Unfortunately, Internet Explorer 4 cannot handle search strings locally, so this trick does not work in HTML Help. Anyway, the bottom topic tries to open the search string topic in it's frameset. When it is loaded, the frame runs the "loaded" script in the help topic. Why have a single frame frameset in the bottom topic? There are two reasons. First, we don't want to have to put the onLoad statement in every annotation page. Second, we don't want to receive an error message if the search string is incorrect.

### How to add annotations

To add an annotation, simply add a new HTML page to the custom directory with "\_custom" added to the filename (for example, "h\_helpfilename\_custom.htm"). JavaScript will show the paperclip and create the link if it finds a match.

### Requirements

The script only has one requirement: a help folder-level htaccess file and a blank.htm file for it to point to, which will contain something like "help topic does not exist." I had to add this file because of a JS bug. If your site has its own 404 error file, and most sites do, JS will report that the requested file is loaded in the hidden frame, even if the 404 error message file is really there. If you do not have a specified error file, then the browser uses a "built in" error message file. Since this file does not "live" on your web server, you cannot read its contents and you get an "access denied" message. The workaround I came up with is to ask for the page's title instead. That works fine. The

htaccess file should contain:

```
DirectoryIndex index.html
ErrorDocument 404 http://www.yourblankpage.com/blank.htm
```

**Note:** Opera shows the hidden frame with the annotation at the bottom of the page. It's not a big problem, but it's not supposed to be visible. Netscape 6 is still a little unstable. It seems to work intermittently with this script.

```
<html>
<head>
<title>Supporting client annotations</title>
</head>

<SCRIPT LANGUAGE="JavaScript1.1">
// (c) 2001 Scott DeLoach, User First Services
onerror = iefix;
noteurl =
location.href.substring(location.href.lastIndexOf("/") + 1,
location.href.length - 4) + "_custom.htm";

top.fbottom.document.location = "bottom.htm?" + noteurl;

function loaded() {
if (top.fbottom.note.document.location.title != "no notes") {
if (navigator.appVersion.indexOf("MSIE") == -1)
document.apaperclip.visibility = "show";
if (navigator.appVersion.indexOf("MSIE") != -1)
document.all.apaperclip.style.visibility = "visible";
}
}

function iefix(msg, URL, linenum) {
note = "404 Not Found";
document.all.apaperclip.style.visibility = "hidden";
return false;
}

function shownote() {
url = top.fbottom.note.document.location.href;
window.open(url, 'helpwin', 'toolbar=0,location=0, ↵
directories=0,status=0,menubar=0,scrollbars=0,resizable=0, ↵
width=520,height=350');
}
</script>

<body>
<p><b>Sample page with a note</b></p>
<p>&nbsp;</p>

<div id="apaperclip" style="position:absolute; width:576px;
height:28px; z-index:2; visibility: hidden">
  <a href="javascript:shownote()"></a>
  click the paperclip to see the note (and no, you don't have
to use a paperclip icon) :-> </div>

<p><a href="note_no.htm">show a page without a note</a></p>

</BODY>
</HTML>
```

## Showing hidden text before printing

---

- IE 4
- IE 5
  
- Netscape 4
- Netscape 6
  
- Opera 5
  
- HTML Help

This script allows a print button to show all hidden text, print it, and then hide it again. The script is called from an onClick event in a link. The script finds any layers and checks their visibility. If they are currently hidden, they need to be shown, printed, then hidden again. The hidden layers' ids are stored in a variable (called dumbo, since elephants don't forget). When the printing is complete, the window receives the focus again. I added an onFocus event to the BODY tag to hide the previously hidden layers. The onFocus event runs the printhide() function, which hides any layer who's id is found in dumbo.

Since Internet Explorer 4 does not support the window.print() command and there's not a great way to tell IE4 from IE5, I use VBScript to print through a security hole in Internet Explorer. Please don't tell Microsoft. ☺

**Note:** Opera doesn't support window.print().

```
<html>
<head>
<title>Printing hidden information</title>
<script>
// (c) 2001 Scott DeLoach, User First Services
br = "net";
if (document.all) br="ie";
printing = 0;
dumbo = "";

function printit() {
if (br == "ie") {
    for (var i=0; i<document.all.length; i++) {
        if (document.all(i).id != "") {
            if (document.all(i).style.visibility == "visible")
dumbo = dumbo + document.all(i).id;
            document.all(i).style.visibility = "visible";
        }
    }
}

if (br == "net") {
    for (var i=0; i<document.layers.length; i++) {
        if (document.layers[i].id != "") {
            if (document.layers[i].visibility == "show") dumbo =
dumbo + document.layers[i].id;
            document.layers[i].visibility = "show";
        }
    }
}
if (br == "net") { window.print(); }
printing = 1;
}

function printhide() {
if (printing != 0) {
if (br == "net") {
    for (var i=0; i<document.layers.length; i++) {
        if (document.layers[i].id != "") {
            if (dumbo.indexOf(document.layers[i].id) == -1)
document.layers[i].visibility = "hide";
        }
    }
}
}
```



```

if (br == "ie") {
    for (var i=0; i<document.all.length; i++) {
        if (document.all(i).id != "") {
            if (dumbo.indexOf(document.all(i).id) == -1)
document.all(i).style.visibility = "hidden";
        }
    }
}
printing = 0;
}
}
</script>
</head>

<body bgcolor="#FFFFFF" onFocus="printhide()" >
<h1>Print hidden Information</h1>
<a href="#" onClick="printit();window.print();">print all</a>
<p>&nbsp;</p>
<p><b>Section 1</b></p>
<div id="Layer1" style="position:absolute; width:70px;
height:24px; z-index:1; visibility: hidden">something
</div>
<p>&nbsp;</p>
<p><b>Section 2</b></p>
<div id="Layer2" style="position:absolute; width:101px;
height:27px; z-index:2; visibility: hidden">something else
</div>
<p>&nbsp;</p>
<p><b>Section 3</b></p>
<div id="Layer3" style="position:absolute; width:138px;
height:26px; z-index:3; visibility: hidden">something or
another </div>

<script language="JavaScript"><!--
DA = (document.all) ? 1 : 0

function handle_error()
{ return true; }
</script>

<script language="VBScript">
sub window_onunload
    on error resume next
    set WB = nothing
end sub

sub print

    OLECMDID_PRINT = 6
    OLECMDEXECHOPT_DONTPROMPTUSER = 2
    OLECMDEXECHOPT_PROMPTUSER = 1

    on error resume next

    'IE4 object has different command structure
    if DA then
        call WB.ExecWB(OLECMDID_PRINT, OLECMDEXECHOPT_PROMPTUSER)

    else
        call WB.IOleCommandTarget.Exec(OLECMDID_PRINT
,OLECMDEXECHOPT_PROMPTUSER,"","")

    end if

```

```

        if err.number <> 0 then
            if DA then ' IE4 they probably cancelled
            '
                alert "Nothing Printed :" & err.number & " : " &
err.description
            else
                handle_error ' ie3x give alternate instructions
            end if
        end if
    end sub

if DA then
    'this must be IE4 or greater
    wbvers="8856F961-340A-11D0-A96B-00C04FD705A2"
else
    'this must be IE3.x
    wbvers="EAB22AC3-30C1-11CF-A7EB-0000C05BAE0B"
end if

document.write "<OBJECT ID=""WB"" WIDTH=0 HEIGHT=0
CLASSID=""CLSID:"
document.write wbvers & """"> </OBJECT>"
</script>

</body>
</html>

```

## Closing popups when they lose focus or if Esc is pressed

---

- IE 4
- IE 5
- ⊙ Netscape 4
- ⊙ Netscape 6
- Opera 5
- HTML Help

This script closes a popup window when the user presses the Esc key or when the popup window loses focus. It uses the onBlur event to close the window when it loses focus and the onKeyPress event to close the window if the Esc key is pressed.

**Note:** Netscape can only capture keypresses within a text field. However, Netscape will close a window when it loses focus. Opera does not support the onKeyPress event, and it will not close a window when it loses the focus.

```

<html>
<head>
<title>Closing popup window with Esc</title>

<script>
function closehelp() {
var key = event.keyCode;
if (key == 27) window.close();
}
</script>

</head>

<body onBlur="javascript:window.close()"
onKeyPress="closehelp()">

Press Esc to close this topic.

</body>
</html>

```

- IE 4
- IE 5
- Netscape 4
- Netscape 6
- Opera 5
- HTML Help

## Closing all help windows when the main help window closes

This script closes all of your help windows when the main help window closes. When the main window is closed, the onUnload event checks to see if the "secondhelp" window is open. If so, it is closed.

```
<html>
<head>
<title>Closing all help windows when the main help window
closes</title>

<script language="JavaScript">
function opener() {
secondhelp = window.open('example.htm','windowname', ↵
'toolbar=0,location=0,directories=0,status=0,menubar=0, ↵
scrollbars=0,resizable=0,width=520,height=350');
}
</script>

</head>

<body onUnload="if (secondhelp && !secondhelp.closed)
secondhelp.close()">

<a href="javascript:opener()">open secondary window</a>

</body>
</html>
```

## Help!

If you have not used JavaScript before, you might be a little overwhelmed by some of these scripts. I'd recommend the following books if you are interested in learning more:

*Designing with JavaScript*  
Nick Heinle

*JavaScript Complete*  
Steven Holzner

*JavaScript Bible*  
Danny Goodman

Feel free to contact me with questions or comments about these scripts. If you have an idea for a script, send me an email.

**Thanks!**

Scott DeLoach  
Founding Partner, User First Services, Inc.  
404.520.0003  
scott@userfirst.net