Making a Framed Web Site Bookmark Friendly

A Solution Using ASP and VBScript

by Carl Dodson

Frames have been a controversial technology since their inception. One of the most widely cited problems with frames is the difficulty they create for users wishing to bookmark a page. There should be an easy way to bookmark a framed Web page.

At the crux of the frames-bookmark problem is the technology that drives frames. A set of <**FRAMESET**> tags tells the browser how to lay out the frames and what documents to place into them. These <**FRAMESET**> tags are usually embedded on a page without any other HTML code (we are ignoring the <**NOFRAMES**> tags for this article).

When the browser opens the <**FRAMESET**> page, it interprets the <**FRAMESET**> tags, draws the frames, and loads the appropriate documents. The URL that the browser "knows" about hasn't changed - it is still seeing the <**FRAMESET**> page as the current page. Since hyperlinks on a framed Web site are usually targeted to a specific frame, the URL in the Address box never changes - it always points to the page that held the <**FRAMESET**> tags. So this is the page that will get bookmarked.

There are a few ways around this, but most involve complex client-side script, ActiveX controls, or Java applets. I wanted an easier way to solve this problem, and I wanted to be able to add it to an existing framed Web site with a minimum of trouble. Since most of our Web sites are adapted to Active Server Pages (ASP), I looked to ASP first for a solution.

Microsoft Internet Information Server (IIS), version 4.0, comes with two built-in ways to create semi-permanent variables on a Web site. The Application object allows you to declare and initialize variables with global scope when the Web site is started. Their global scope persists until the Web site is stopped, or the object is implicitly destroyed. The Session object is similar, except that it is created with each new session (each new user) and is destroyed when the session ends (by default, 20 minutes after the user's last request to the Web site). These objects are declared in the Global.asa file.

For this article, we only need the Session object . This article assumes that you are using only **one** Global.asa file, and that it is located in the root directory of the Web site. Also, because all of the example code in this article is written in VBScript, some familiarity with VBScript and ASP will be helpful to you.

Here is the Global.asa file for this solution:

```
<SCRIPT Language="VBScript" RUNAT="Server">

Sub Session_OnStart
    '// These variables track the pages currently
    '//  loaded into the frames.
    Session("TOC") = "toc.html"
    Session("MAIN") = "main.html"
End Sub

</SCRIPT>
```

We declare two variables here. **TOC** is the Table of Contents frame, and **MAIN** is the main viewing frame. They are initialized to the default pages that will be loaded when the frames are first drawn. Technically, this isn't necessary (the code that draws the framesets will also initialize the defaults) but it is good coding practice not to make assumptions when the content (toc.html and main.html) is being referenced.

The next step is to lay out the frames themselves. The key to solving the bookmark problem lies in how we represent the <**FRAMESET**> tags.

Since the browser wants a physical URL to point to, we must give it one. We also must ensure that WE control what that URL is at any time. Therefore, we can't just embed <**FRAMESET**> tags on a page. We must generate these tags using ASP.

Like many Web sites that use frames, mine will lay the frames out from the root of the Web site. The page that draws the frames is called Default.asp.

```
<%@ Language="VBScript" %>

<html>
<head>
    <title>
        Using Bookmarks on Framed Web Sites
    </title>
</head>



<%
'// Test for uninitialized variables.
'// If the Session objects are empty,
'// then set them to the default pages.
If Request.QueryString = "" Then
    Session("TOC") = "toc.html"
    Session("MAIN") = "main.html"
Else
'// The QueryString container object holds
'// the queries from hyperlink. Set the
'// Session variables to their respective
'// page.
    Session("TOC") = Request.QueryString("toc")
    Session("MAIN") = Request.QueryString("main")
End If

'// Write out the FRAMESET tags.
Response.Write "<frameset cols='100,*'>"
Response.Write "<frame name='toc' src='" &_
       Session("TOC") &_
       "' scrolling='Auto' FRAMEBORDER='No' noresize>"
Response.Write "<frame name='main' src='" &_
       Session("MAIN") & "' scrolling='auto'>"
Response.Write "</frameset>"

%>

<body>
</body>
</html>
```

The secret to this solution is the Response.Write commands. In the first few lines, we see that the hyperlinks we use on this site must all point to the Default.asp page and pass a set of queries, so that our <**FRAMESET**> tags know what documents to load. The Response.Write commands are constructed with the Session variables. If you were to point your browser to this Default.asp page and then view the source in the browser, this is what you would see:

```
<html>
<head>
    <title>
        Using Bookmarks on Framed Web Sites
    </title>
</head>

<frameset cols='100,*'>
    <frame name='toc' src='toc.html' scrolling='Auto'
    FRAMEBORDER='No' noresize><frame name='main'
    src='main.html' scrolling='auto'>
</frameset>

<body>
</body>
</html>
```

The browser has a real URL to point to. If you bookmark this page, the bookmark will work as expected.

However, we want ***any*** page to be able to be bookmarked. Let's take a look at the toc.html page and see how that's accomplished.

```
<html>
<head>
    <title>
        Using Bookmarks on Framed Web Sites
    </title>
</head>

<body>


<br><br>

<a href="default.asp?toc=toc.html&main=main.html"
    target="_top">Selection 1</a>
<br><br>
<a href="default.asp?toc=toc.html&main=main2.html"
    target="_top">Selection 2</a>

</body>
</html>
```

Notice that the hyperlinks contain query variables. We pass a variable named **toc**, which contains the string "toc.html". The **main** variable contains the string "main.html" or "main2.html", depending on which hyperlink is clicked.

There are clearly drawbacks to this solution. First, the browser's Address box now shows a fresh set of queries every time you click a hyperlink. For visitors who are used to seeing a real page name in the Address box, this may be a bit irritating. Also, we are violating one of the reasons for having frames in the first place - the Table of Contents page is refreshed with every link. If large graphics or excessive content make up one of the refreshing frames (like our TOC frame), then we lose some of the speed advantage of frames. This is offset somewhat if the visitor has caching enabled, but the problem still exists.

There are other ways to enable bookmarks in frames, but they tend to be more complex. Some are browser-specific, which limits who can visit your Web site. The solution offered here is not the most elegant, but it very easy to adapt to existing Web sites, and it is very flexible. Use this solution as a foundation for more comprehensive Web applications.

Happy Coding!

Developed Under:
Visual InterDev 6.0 Service Pack 4, ASP 3.0