


[MSDN Home](#)
[Developer Centers](#)
[Library](#)
[Downloads](#)
[How to Buy](#)
[Subscribers](#)
[Worldwide](#)

Search for



[Advanced Search](#)
[MSDN Magazine Home](#)
[December 2004](#)
[Search](#)
[Source Code](#)
[Back Issue Archive](#)
[Column Archive](#)
[RSS](#)
[Subscribe](#)
[Order Back Issues](#)
[Reader Services](#)
[Meet the Staff](#)
[Meet the Authors](#)
[Media Kit](#)
[Special CD and DVD Offers](#)
[Podcast](#)
[Submit an Article](#)
[Write to Us](#)
[Corrections](#)
[TechNet Magazine](#)


## MSDN® Magazine

The Microsoft Journal for Developers

[MSDN Home](#) > [MSDN Magazine](#) > [December 2004](#)

# WEB Q&A

Windowed and Windowless Elements, Cookie Characters, and More

[Edited by Nancy Mitchell](#)


**Q** I have a windowless JavaScript menu that gets overshadowed by a windowed dropdown list. How can I prevent this?

**A** All windowed elements paint themselves on top of all windowless elements, regardless of what their container wants. However, windowed elements do follow the z-index attribute with respect to each other, just as windowless elements follow the z-index attribute with respect to each other. See [How the Z-index Attribute Works for HTML Elements](#), for more on how this works.

So, one way to deal with this is to convert the menu to a DHTML scriptlet and place this <OBJECT> into a floating <DIV>, like so:

```
<div id='ctl00_MasterContentplaceholder_newdate_div'
    style='position: absolute; display: inline; z-index: 8;' class=''>
  <OBJECT ID='ctl00_MasterContentplaceholder_newdate_objj'
    DATA=' /aspnet_client/esupplyctl/ClientTools2.htm' style='z-index: 8'
    TYPE='text/x-scriptlet' >
  </OBJECT>
</div>
```

An overview of scriptlets is available at [Introduction to Scriptlets](#). Plus, the book Instant Scriptlets by Dino Esposito (Wrox, 1998) has lots of information on creating DHTML scriptlets, if you need further details on how to do this.

Another option is to use window.createPopup for your menu, which is also a windowed element. You can use an HTML Component (HTC) and an XML file defining the menu items.

**Q** I have been getting the following cookie-related error from an ASP.NET page: "A potentially dangerous Request.Cookies value was detected from the client: (W0073355\_ID="..icVVWBPvb90np1oMFqAQ=")".

It looks like some combination of characters is raising this exception. How can I change this behavior? There is no need to flag this cookie since there isn't any text after the equal signs.

**A** As you know, this is a security feature designed to prevent script from being submitted with an HTML form. Although none of the base 64 characters are invalid in a cookie value, there are combinations that are not acceptable, including:

- <{ a-z }
- <!
- expression(
- on { a-z } \* =
- &#
- script{ space} \*.

If the HttpRequestValidationException is being raised, then the only way to work around it is to disable request validation for the pages using those cookies. However, if you disable request validation, it is very important that you perform your own validation of any input accepted from the user. The request validation feature, which was introduced in ASP.NET 1.1, is designed to be a first line of defense against cross-site scripting attacks. Without such validation, your application could be vulnerable to such attacks.

**Q** When I switch from ASP to ASP.NET does that mean I must move to a two-tier architecture from a three-tier architecture?

**A** There aren't any changes that would limit the number of tiers in your ASP.NET application. That is really a design decision. So no, switching to ASP.NET does not mean that a three-tier architecture must be turned into a two-tier version.

Actually, ASP.NET allows more tiers because you can separate the presentation code (.aspx) completely from the page logic (a .cs or a compiled class in a .dll). At that point you can choose to put your business logic in your .dll in the bin folder, or in another separate .dll, or even in the stored procedures in SQL Server™. You can have as many tiers as you want.

Note that you may want to make sure that you're clear on whether you're talking about physical or logical tiers, however. You can have a three-tier logical architecture while still maintaining a two-tier physical architecture (in order to avoid the cost of cross-process calls between the presentation and business tiers). This is true for both ASP and ASP.NET, but it's an important distinction that is sometimes lost when discussing n-tier architecture.

**Q** I have a query that returns data to a DataGrid depending upon which items are selected in the dropdown lists. The user can then page through the DataGrid to find the information she wants.

The problem is that when the user clicks on a row in the DataGrid it brings her to a new page, and when she clicks to return to the page with the DataGrid, the dropdowns and DataGrid are back to the original state and the user must re-query, which I don't want her to have to do.

**A** The best way to deal with multiple queries to the same data is to cache the DataSet in the ASP.NET cache. You can set the cache expiration policy so that it's appropriate for your data (a short expiration period if your data changes a lot, long or non-expiring if your data rarely changes). Then when you need to bind your data to the DataGrid, you can retrieve the DataSet from the cache if it's still there (memory pressure can expire data from cache) or re-query your data if it's not. The Knowledge Base article 323290 has a good example under the heading "[Cache arbitrary objects in server memory](#)".

**Q** I need to package a control I've built into a Microsoft® Installer (MSI) file that will include a number of DLLs, and I need to have the installer run on a user's machine when he visits my Web site. Is there anything special that I need to do to cause the OCX file to be installed and registered to execute with Microsoft Internet Explorer? Can I use the <OBJECT> tag so the user does not have to click on a link?

**A** For this to work in Internet Explorer, you just need to register the file like any other ActiveX® control to be used with an <OBJECT> tag. That means that in the MSI, the Class table needs to be filled out along with the Typelib table. As an alternative solution, you can use the SelfReg table.

The control itself should support IProvideClassInfo or IProvideClassInfo2 for any event handling via the <OBJECT> tag. The Knowledge Base article 200839 ("[How To Enable ActiveX Control Event Handling On A Web Page](#)") provides a further explanation.

It is acceptable to use the <OBJECT> tag if this is outside of the MSI, but users are still going to get prompts due to security constraints. You might consider just having a script redirect to the setup page if the control is not on the machine, using code like that shown in [Figure 1](#).

There is a document on MSDN® that discusses no-touch deployment in the Microsoft .NET Framework, but that solution also requires end-user configuration. (See [No-Touch Deployment in the .NET Framework](#).)

**Q** I have an ASP.NET application that uses the Internet Explorer Web controls. Each time the user navigates to a new page or does a postback to the same page, the HTC files are retrieved again from the server. I set the Content expiration for the HTC files on the server, but Internet Explorer does not seem to cache them. Is there anything else I should be doing?

**A** A supported hotfix is now available from Microsoft, but it is only intended to correct the problem that is described in the Knowledge Base article 840312 ("[Cached HTC Files Generate Additional GET Requests In Internet Explorer 6](#)"). You should only apply it to systems that are experiencing this specific problem. Therefore, if you are not severely affected by this problem, Microsoft recommends that you wait for the next Internet Explorer 6.0 service pack that contains this hotfix.

**Q** I'm trying to troubleshoot Web connection problems and want to look at headers. Where can I find a tool to display HTTP headers in Internet Explorer?

**A** When you are troubleshooting connectivity issues between IIS and Web clients, you may want to view data that is not displayed in the Web browser, such as the HTTP headers that are included in the Request and Response packets. One way is to use wfetch. It's a free utility that comes with the IIS 6.0 Resource Kit and is also available as a standalone app.

You can also use wfetch to provide detailed information about the traffic between the client and server. See the Knowledge Base article 284285 ("[Use Wfetch.exe To Troubleshoot HTTP Connections](#)").

While wfetch is a great tool for some kinds of debugging, if you need a tool that integrates with Internet Explorer so you can surf and look at the headers at the same time, check out [ieHTTPheaders](#).

**Q** I have a page that shows a DataGrid based on user filters. When the user clicks on an Edit button for a particular row in the DataGrid, I use Response.Redirect to redirect the user to the details page where he updates the data. I now want to take my user to the first page and show him the DataGrid based on the same filters he had selected earlier on. What is the best way to do this? My other requirements are as follows:

- I want to preserve some values in Page1.aspx, go to Transfer.aspx, and then when I come back to Page1.aspx I want to use those values.
- I need to use query strings to pass values from Page1.aspx to Transfer.aspx and back.
- Finally, I'd like to catch the URL of the calling page. I also have to consider performance.

I know I can store filter selections in session state variables and when the user comes back to the page, the DataGrid can use these session variables. However, session variables depend on cookies, so if the user doesn't have cookies enabled, I have a problem.

**A** One way to solve your problem is to use Server.Transfer and Context.Items. Here's the code you would use in Page1.aspx:

```
Context.Items.Add("pageval", "value to preserve");
Server.Transfer("Transfer.aspx");
```

And in Transfer.aspx you would use:

```
string val = (string)Context.Items["pageval"];
```

Server.Transfer performs a transfer on the server side; the page does not go back to the client in between, so this approach would avoid the 302 page sent to the client which results in an extra round-trip between the client and server.

The HttpContext instance for a request is available while the processing occurs on the server side. The moment that the page is sent to the client, that particular HttpContext is no longer valid. It is not preserved between round-trips to the client. So, if you go to Transfer.aspx and use Server.Transfer to go back to Page1.aspx, the context would still be available. If you let the client see Transfer.aspx, then it would not be available.

Yes, you can use a query string to pass values as well. You can use this method with Server.Transfer and Response.Redirect, which is a good way to pass values back and forth. With Server.Transfer, the client will actually never see the query string, but they will see it with Response.Redirect. There are also ways to encode your values inside the query string so that critical information can be encrypted.

There's one more thing you should know about Server.Transfer. In ASP.NET 1.0, the transfer would be preserved as a POST if that was the initial request. In ASP.NET 1.1, unless the page is transferred to itself, it will always be a GET, so you would have to use the query string because the Form collection is not available anymore.

To catch the URL of the calling page you can choose to use either Request.UrlReferrer.AbsoluteUri (which gets the referring page path and query string) or Request.UrlReferrer.AbsolutePath (which gets just the referring page path).

In the page Load event of Page1.aspx, Request.UrlReferrer will be null because there is no referrer the first time a user types the URL of this page. Thus, any attempts to access properties such as AbsoluteUri will result in an exception. You can avoid this problem by only executing this code in Page1.aspx if Page.IsPostBack is true. You can use something like this:

```
string referrerUri = (Request.UrlReferrer != null) ?
    Request.UrlReferrer.AbsoluteUri : string.Empty;
```

The only real problem with using UrlReferrer is that it is not guaranteed to be there even when the page is actually being referred to because some firewalls and privacy software filter out that value. You can tack a query string parameter like "ReferrerUrl=MyPageName.aspx" onto the end of your Response.Redirect call.

For more information, visit [HttpServerUtility.Transfer Method](#), and [HttpContext Class](#).

Got a question? Send questions and comments to [webqa@microsoft.com](mailto:webqa@microsoft.com).

Thanks to the following Microsoft developers for their technical expertise: Gaurav Bansal, Seth Behunin, Daniel Boerner, Todd Carter, Richard Ciapala, G. Andrew Duthie, Rob Franco, Scott Golubenko, Bac Hoang, Kenneth Lassen, Pat Long, Alexander Louie, David Lovell, Nathan Manis, Oscar Newkerk, Polita Paulus, Ranjit Sawant, Ivan S. Shestakov, Martin Sieber, and John Wadie.



From the [December 2004](#) issue of [MSDN Magazine](#).

[Back to top](#)

QJ: 041208

© 2007 Microsoft Corporation and CMP Media, LLC. All rights reserved; reproduction in part or in whole without permission is prohibited.

Software Rights Management:  
**Want Multiple Options?**



**Aladdin**  
SECURING THE GLOBAL VILLAGE  
[www.Aladdin.com](http://www.Aladdin.com)

**RESOURCES**

**Featured Topic**

SharePoint  
Development:  
Four Objects That  
You Need To Know



Addison Wesley | **SAMS**




**Additional Topics**

More resources from  
[msdnmag@resources.com](mailto:msdnmag@resources.com)

**INTEROP**  
New York  
Oct. 22-26, 2007

**REGISTER TODAY**



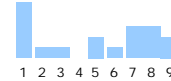
 Print  E-Mail  Add to Favorites

How would you rate the quality of this content?

1 2 3 4 5 6 7 8 9  
Poor          Outstanding

Tell us why you rated the content this way. (optional)

Average rating:  
5 out of 9



18 people have rated this page

[Manage Your Profile](#) | [Legal](#) | [Contact us](#) | [MSDN Flash Newsletter](#)

**Microsoft**

© 2007 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

Requirements 2004/11/18