**Response.Redirect Method**

**IIS 6.0**

The [Redirect](http://msdn.microsoft.com/en-us/library/ms524309(v=vs.90).aspx) method causes the browser to redirect the client to a different URL.

Redirect(

URL

)

[Parameters](javascript:void(0))

URL

The Uniform Resource Locator (URL) that the browser is redirected to. This can be an full URL beginning with "http://", a virtual path to a location on the same IIS server, or the name of a file contained in the same location as the original URL.

URL can include a query string.

Older Web browsers might convert a POST request to a GET request during a redirection.

|  |
| --- |
| **Caution noteCaution:** |
| Always validate and encode the URL that is passed to Response.Redirect to protect against cross-site scripting attacks. For information about how to remove harmful characters from strings, see [Removing Harmful Characters from User Input](http://msdn.microsoft.com/en-us/library/ms526004(v=vs.90).aspx). |

[Return Values](javascript:void(0))

This method has no return values.

[Applies To](javascript:void(0))

[Response Object](http://msdn.microsoft.com/en-us/library/ms525405(v=vs.90).aspx)

[Remarks](javascript:void(0))

Any response body content such as displayed HTML text or Response.Write text in the page indicated by the original URL is ignored. In addition, code execution in the current page is terminated when the Redirect method is processed, so subsequent code in the page will also be ignored.

However, this method does send other HTTP headers set by this page indicated by the original URL to the client. An automatic response body containing the redirect URL as a link is generated. The Redirect method sends the following explicit header, where URL is the value passed to the method, as shown in the following code:

HTTP 1.0 302 Object Moved

Location: http://www.microsoft.com

[Example Code](javascript:void(0))

The following example shows you how to use the VBScript programming language to redirect the user to the Microsoft Web site after validating the URL.

VBScript

<%@ LANGUAGE="VBScript" %>

<%

Dim MyUrl

MyUrl = "http://www.microsoft.com"

Response.CodePage = 1252

If ValidateInput(MyUrl) Then

Response.Redirect (myURL)

Else

Response.Write("URL was invalid.")

End If

Function ValidateInput(sInput)

Dim reValid

Set reValid = New RegExp

reValid.Pattern = "^[\w\.:\?&=/]\*$"

reValid.MultiLine = False

reValid.Global = True

ValidateInput = reValid.Test(sInput)

End Function

%>

The following example shows you how to use the VBScript programming language to redirect the user to a virtual directory on the same IIS server.

VBScript

<% Response.Redirect "/samples/asp/newpage.asp" %>

The following example shows you how to use the VBScript programming language to redirect the user to a local file while passing a query string.

VBScript

<% Response.Redirect Server.HTMLEncode("newpage.asp?var1=5&var2=7") %>

Form and query string data is not transferred to the new URL. The following example shows you how to use the VBScript programming language to pass a querystring from the original request to the new URL.

VBScript

<%

dim qs

qs = Server.URLEncode(Request.Querystring)

Response.Redirect "newpage.asp?" + Server.HTMLEncode(qs)

%>

# Server.Transfer Method

**IIS 6.0**

The [Transfer](http://msdn.microsoft.com/en-us/library/ms525800(v=vs.90).aspx) method sends all of the information that has been assembled for processing by one .asp file to a second .asp file.

Transfer(

Path

)

[Parameters](javascript:void(0))

Path

The location of the .asp file to which control should be transferred.

[Return Values](javascript:void(0))

This method has no return values.

[Example Code](javascript:void(0))

The following example demonstrates transferring from one .asp file to another, as well as sending the session identifier to the client.

The output from these scripts is:

A session ID

I am going to ASP2

The same session ID

--- ASP1 ---

<HTML><BODY>

<% Dim sessvar1 Response.Write Session.SessionID

Response.Write ("<BR>")

Response.Write("I am going to ASP2 <BR>")

Server.Transfer("/Myasps/ASP2.asp")

%>

--- ASP2 ---

<HTML>

<BODY><% Response.Write Session.SessionID%>

</BODY>

</HTML>

### Applies To

[Server Object](http://msdn.microsoft.com/en-us/library/ms525541(v=vs.90).aspx)

[Remarks](javascript:void(0))

When you use the [Transfer](http://msdn.microsoft.com/en-us/library/ms525800(v=vs.90).aspx) method, the state information for all the built-in objects are included in the transfer. This means that any variables or objects that have been assigned a value in session or application scope are maintained. In addition, all of the current contents for the [Request](http://msdn.microsoft.com/en-us/library/ms524948(v=vs.90).aspx) collections are available to the .asp file that is receiving the transfer.

The [Transfer](http://msdn.microsoft.com/en-us/library/ms525800(v=vs.90).aspx) method returns the ASP 0173 error, "Invalid Path Character", if the Path parameter contains any of the following characters:

* Asterisk (\*)
* Question mark (?)
* Angle brackets (< or >)
* Comma (,)
* Colon or semi-colon (: or ;)
* Single-quote or double-quote (' or ")
* Right square bracket (])
* Double slashes (// or \\)

If the path you specify in the input parameter is for an .asp file in another application, the .asp file executes as if it were in the application that contains the [Server.Transfer](http://msdn.microsoft.com/en-us/library/ms525800(v=vs.90).aspx) command. In other words, all variables and objects that have been given application scope either by other .asp files in the application or by the application's Global.asa file are available to the called .asp file. However, the path parameter must not contain a query string, or ASP returns an error.

[Server.Transfer](http://msdn.microsoft.com/en-us/library/ms525800(v=vs.90).aspx) acts as an efficient replacement for the [Response.Redirect](http://msdn.microsoft.com/en-us/library/ms524309(v=vs.90).aspx) method. [Response.Redirect](http://msdn.microsoft.com/en-us/library/ms524309(v=vs.90).aspx) specifies to the browser to request a different page. Because a redirect forces a new page request, the browser makes two requests to the Web server, so the Web server handles an extra request. IIS 5.0 introduced a new function, [Server.Transfer](http://msdn.microsoft.com/en-us/library/ms525800(v=vs.90).aspx), which transfers execution to a different ASP page on the server. This avoids the extra request, resulting in better overall system performance, as well as a better user experience.