BlueHost Tutorials

Easy, hassle-free Web Hosting for $6.95/mo.
Unlimited Space, Transfer & Domains on 1 Account!
bluehost
Sign Up Now

Patricia Geary
February 2010

DISCLAIMER: The above BlueHost link is an affiliate link.

Table of Contents

How to make a subfolder the main folder for your BlueHost main domain

I recently moved my hosting to BlueHost and have been very happy with both the tech support and their services. BlueHost allows you to host multiple domains within the one account. The main domain on the hosting account uses the public_html folder for all of its web site files. Any addon domains use subfolders inside the public_html folder. When your viewer types in the url or clicks on a link to the url, that viewer is directed to the domain and sees the url in the address bar. No problem so far.

I use Expression Web as my web editor and publish using the ftp facility from within the program in most cases. I had been using Filezilla to publish to the main domain frontpage-to-expression.com as there are two of us who maintain the site. Imagine my surprise the first time I used the ftp client from within Expression Web to have the program compare the domain I was publishing to ALL the folders for every domain within the public_html directory.

There had to be a way to remedy this so I went searching the BlueHost support forums and found it. If you are hosting your domains on BlueHost and do NOT want all of the files for your main domain within the public_html directory as well as all of the addon domain folders you can follow the .htaccess How to make a subfolder the main folder for your main domain from the BlueHost knowledge base or continue with this tutorial below.

Visitors to your Web site will not be able to tell that your main domain is using a subfolder, they will still see the Web site address as http://www.your-domain.com/page.html BUT for SEO purposes I want my visitors to see http://www.your-domain.com/ so I made some changes in the code BlueHost gives for achieving this. You can see the results by visiting http://www.frontpage-to-expression.com/

Creating the .htaccess file

This is the information provided by BlueHost:

```
# Bluehost.com
# .htaccess main domain to subfolder redirect
# Copy and paste the following code into the .htaccess file
# in the public_html folder of your hosting account
# make the changes to the file according to the instructions.

# Do not change this line.

RewriteEngine on

# Change your-domain.com to be your main domain.

RewriteCond %{HTTP_HOST} ^(www.)?your-main-domain.com$

# Change 'subfolder' to be the folder you will use for your main domain.

RewriteCond %{REQUEST_URI} !^/subfolder/
```

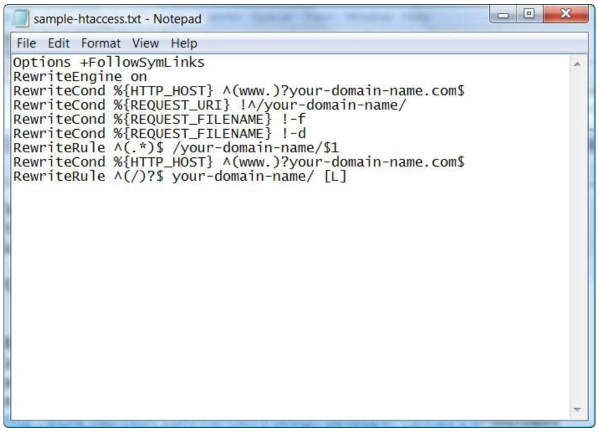
```
# Don’t change this line.

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d

# Change 'subfolder' to be the folder you will use for your main domain.

RewriteRule ^(.*)$ /subfolder/$1

# Change yourdomain.com to be your main domain again.

# Change 'subfolder' to be the folder you will use for your main domain

# followed by / then the main file for your site, index.php, index.html, etc.  NOTE: I do NOT want to be redirected to a specific file name so modified the code here.

RewriteCond %{HTTP_HOST} ^(www.)?yourmaindomain.com$ RewriteRule ^(/)?$ subfolder/index.php [L]
```

My final code looks like this:

```
Options +FollowSymLinks
RewriteEngine on
RewriteCond %{HTTP_HOST} ^(www.)?frontpage-to-expression.com$
RewriteCond %{REQUEST_URI} !^/frontpage-to-expression/
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ /frontpage-to-expression/$1
RewriteCond %{HTTP_HOST} ^(www.)?frontpage-to-expression.com$
RewriteRule ^(/)?$ frontpage-to-expression/ [L]
```

If you wish, you may download the sample-htaccess.txt file and edit for YOUR domain.

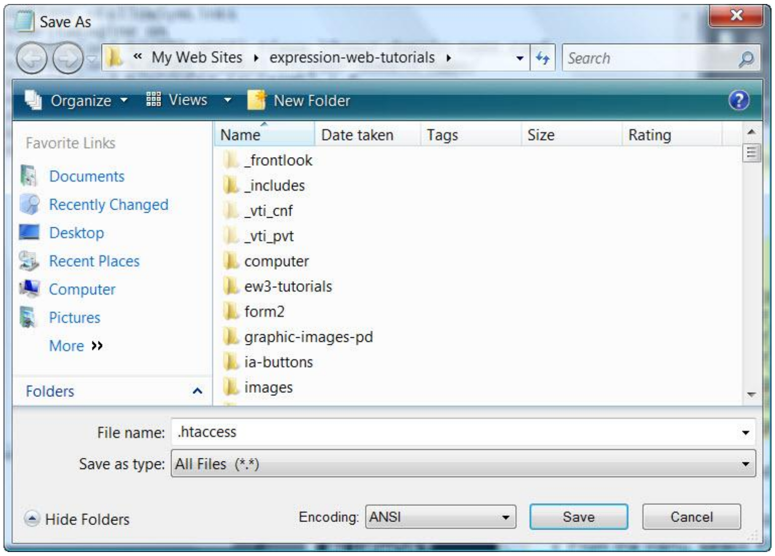
Open the above file in Notepad



```
Options +FollowSymLinks
RewriteEngine on
RewriteCond %{HTTP_HOST} ^(www.)?your-domain-name.com$
RewriteCond %{REQUEST_URI} !^/your-domain-name/
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ /your-domain-name/$1
RewriteCond %{HTTP_HOST} ^(www.)?your-domain-name.com$
RewriteRule ^(/)?$ your-domain-name/ [L]
```

Figure 1 - Sample .htaccess file in Notepad.

- From the menu, select: File > Save As.
- A dialog box will open. At the top of the box make sure it has the folder where your website is located next to the Save in area.
- To the right of File name: type in .htaccess
- To the right of Save as: type, select All Files.



File name: .htaccess

Save as type: All Files (*.*)

Encoding: ANSI

Figure 2 - File > Save as dialog box.

- Click the save button.

Uploading the .htaccess file

Once edited, you will need to upload the .htaccess file using your ftp program. You should have received a welcome email from BlueHost with your temporary information that will look something like this:

Temporary Web Address:

http://22.333.444.555/~username/
Temporary FTP Address:
ftp://username@22.333.444.555/public_html

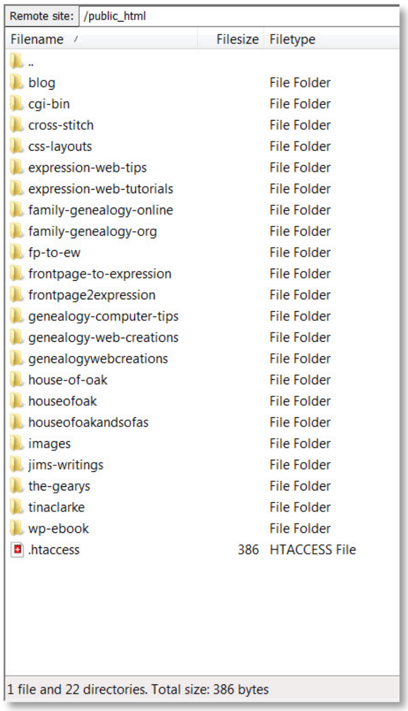
Before you can upload the file you will need to set up your ftp program with the correct settings. Among the various free FTP programs available to download on the internet is FileZilla which I use and highly recommend. It can be downloaded free of charge from http://www.filezilla-project.org. You can access a detailed tutorial from BlueHost Support on Getting Started Wizard :: FileZilla FTP

- Publish the .htaccess to the public_html directory.

- Create a new folder within the public_html directory with the same name you used for the folder when creating the .htaccess file i.e. your-domain.com now points to the folder your-domain.

I also add an .htaccess file to the new folder you just created that will redirect the non www version of the domain to the www version of the domain. See Google Canonical problems - www versus non.www.

Now you have your public_html root directory that contains the newly created .htaccess file and subfolders for each of your domains not just your addon domains.



Remote site: /public_html

Filename	Filesize	Filetype
..		
blog		File Folder
cgi-bin		File Folder
cross-stitch		File Folder
css-layouts		File Folder
expression-web-tips		File Folder
expression-web-tutorials		File Folder
family-genealogy-online		File Folder
family-genealogy-org		File Folder
fp-to-ew		File Folder
frontpage-to-expression		File Folder
frontpage2expression		File Folder
genealogy-computer-tips		File Folder
genealogy-web-creations		File Folder
genealogywebcreations		File Folder
house-of-oak		File Folder
houseofoak		File Folder
houseofoakandsofas		File Folder
images		File Folder
jims-writings		File Folder
the-gearys		File Folder
tinaclarke		File Folder
wp-ebook		File Folder
.htaccess	386	HTACCESS File

1 file and 22 directories. Total size: 386 bytes

Figure 3 - Root directory on Bluehost server.

Now that you have created your new main domain directory and uploaded your .htaccess file, you are ready to publish to the newly created directory. Additional files created by BlueHost when the main domain is setup can also be moved to the newly created directory. They are various error pages - 400.shtml, 404.shtml, etc. These pages can be edited through your Control Panel OR your web editor.

Custom Error Pages - Customizing BlueHost Error Pages

Once you set up your domain on your BlueHost account, error pages are added to the domain directory. All are named with the .shtml file extension. They include:

- 400 (Bad Request)
- 401 (Authorization Required)
- 403 (Forbidden)
- 404 (Not Found)
- 500 (Internal Server Error)

You can choose to leave these pages as they are, edit them through the Control Panel, OR, the method I prefer, editing them with your web editor.

To Customize your 'Error Pages' Using the Control Panel

Step 1: Log into your Bluehost control panel.

Step 2: Locate the **'Advanced'** menu.

Step 3: Locate and click on the Error Pages icon. This will take you to the 'Error pages' menu.

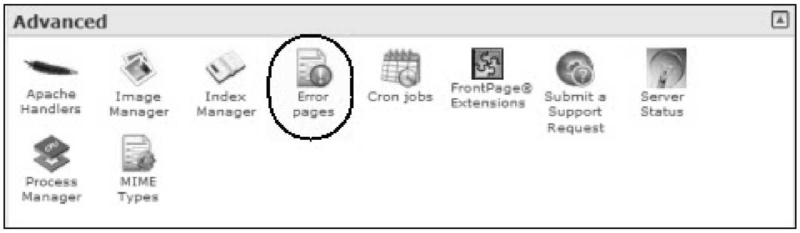
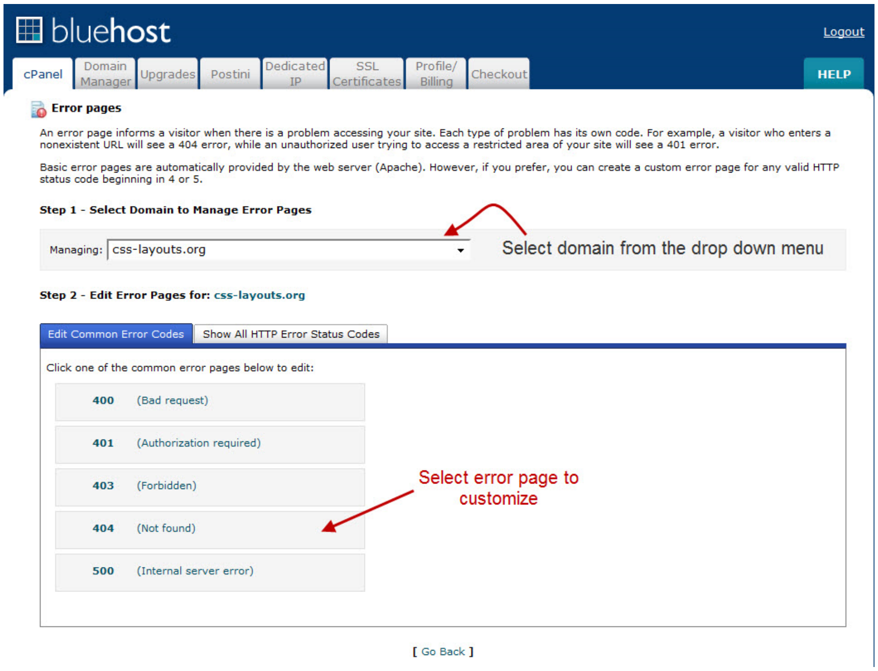


Figure 4 - BlueHost Advanced Menu

Step 4: Select the domain to customize error pages for from the drop down menu. Select he error page to customize.



Error pages

An error page informs a visitor when there is a problem accessing your site. Each type of problem has its own code. For example, a visitor who enters a nonexistent URL will see a 404 error, while an unauthorized user trying to access a restricted area of your site will see a 401 error.

Basic error pages are automatically provided by the web server (Apache). However, if you prefer, you can create a custom error page for any valid HTTP status code beginning in 4 or 5.

Step 1 - Select Domain to Manage Error Pages

Managing: css-layouts.org

Select domain from the drop down menu

Step 2 - Edit Error Pages for: css-layouts.org

Click one of the common error pages below to edit:

400	(Bad request)
401	(Authorization required)
403	(Forbidden)
404	(Not found)
500	(Internal server error)

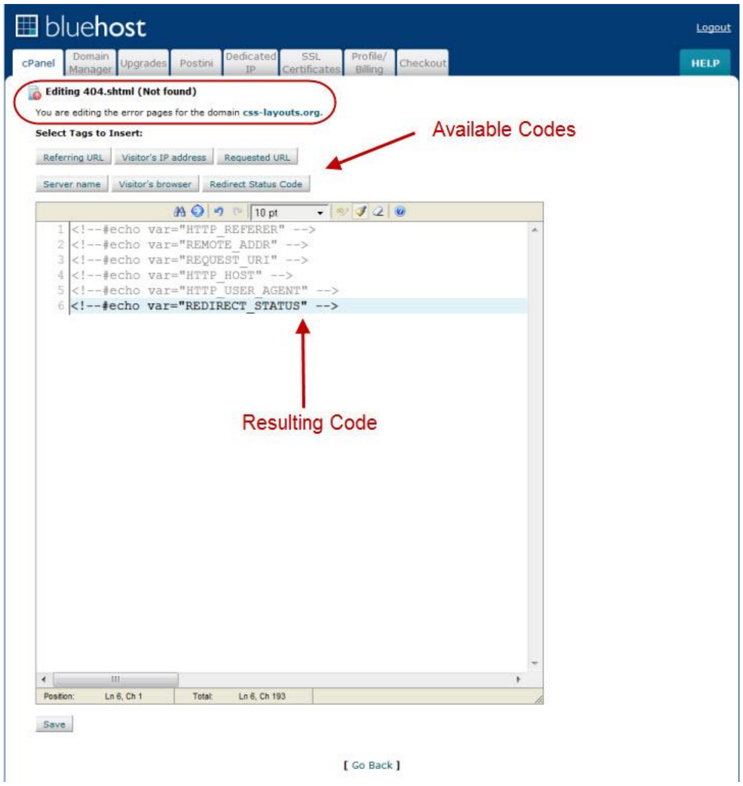
Select error page to customize

Figure 5 - Bluehost Control Panel - Select error page.

Step 5: To customize the error page, you can click on the error page number (400, 401, 403, 404, 500) or the error page name (Bad Request, Authorization required, etc) When the selected Error Page loads, the following six tags are available:

1. Referring URL - Displays the url of the web site the visitor was previously viewing.
2. Visitor's IP Address - Displays the IP address of the visitor viewing the error page.
3. Requested URL - Displays the intended url the visitor is trying to view.
4. Server name - Displays the web site's server name.
5. Visitor's browser - Displays the visitor's browser type such as IE, FireFox, Netscape, etc.
6. Redirect Status Code - This displays the type of code in the 300-307 range.

You do not have to use any of the above tags. They are available to you based on your own personal preferences and needs, specific to your web site. You can place any html code or text in the error page.



Editing 404.shtml (Not found)

You are editing the error pages for the domain css-layouts.org.

```
<!--#echo var="HTTP_REFERER" -->
<!--#echo var="REMOTE_ADDR" -->
<!--#echo var="REQUEST_URI" -->
<!--#echo var="HTTP_HOST" -->
<!--#echo var="HTTP_USER_AGENT" -->
<!--#echo var="REDIRECT_STATUS" -->
```

Figure 6 - Tags available for use on error pages on Bluehost servers.

Tip: You can find what error codes are being shown to your visitors by viewing any of the available web site statistical analysis programs built in to cPanel such as Awstats, Webalizer, etc.

Step 6: If you are satisfied with your customized error page click on the 'Save' button to store your changes. You will be taken to the 'Error page saved' menu.

Step 7: Click the 'Go Back' button to return to the Error pages main menu to edit a different Error Page or Click the 'Home' icon in the upper left hand corner to return to the main cPanel menu.

COMMENT: The resulting error page does NOT give much in the way of information. I prefer to design my own error page and publish it to the server.

To Customize your 'Error Pages' Using Expression Web

You can either create a new page from scratch in Expression Web OR create a new page using your dynamic web template. If you have your pages located in subdirectories as well as your main directory, you will need to change your references to absolute urls (full url) for the error page to work in the subdirectories.

Example: http://www.expression-web-tutorials.com/404.shtml

Example: http://www.expression-web-tutorials.com/403.shtml

Step 1: File > New > Page OR File > New > Create From Dynamic Web Template

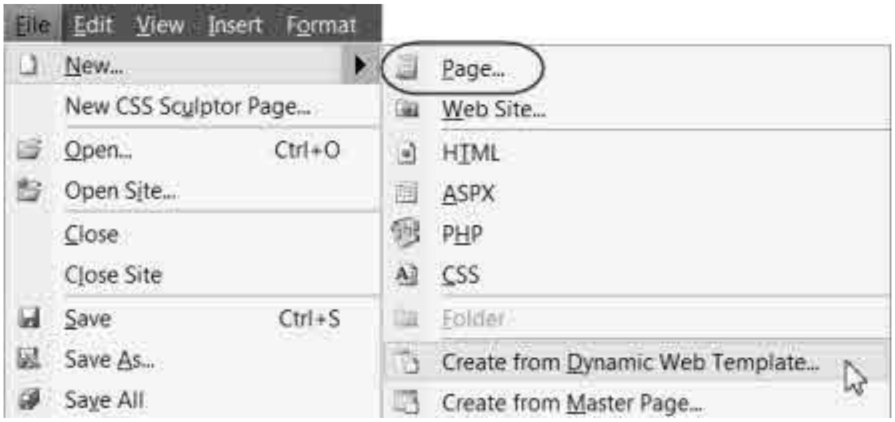


Figure 7 - File > New > Page OR DWT.

Step 2: Save the newly created page with the same name as the error file on the server, i.e., 404.shtml **NOTE:** If you choose to use your DWT, **immediately detach the page from the DWT**. Format > Dynamic Web Template > Detach from Dynamic Web Template.

Step 3: Change all links to style sheets and images to absolute URL's if you want your page to work across the entire site. Edit the page as you wish.

Step 4: Save the error page and publish to your BlueHost server space.

It's as simple as that. Now instead of a blank page or a page that gives no information on your site, you have your own custom error page.

NOTE: The graphic image used on the 403.shtml page is one I purchased and MAY NOT be copied. The image used on the 404.shtml page is one I created and you are free to us.

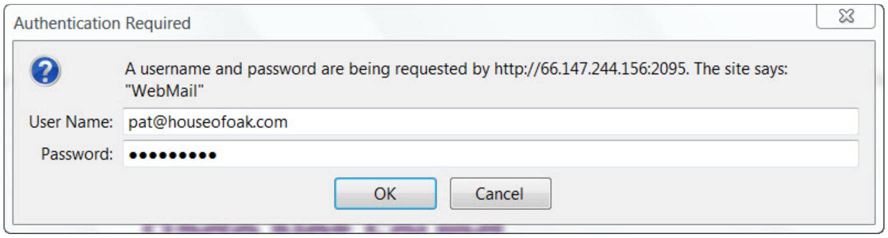
Additional Resources

- 404 Best Practices by CSS Tricks
- 404 Error by S. R. Emerson
- HTTP 404 by Wikipedia
- http://www.404errorpages.com/

BlueHost Webmail

This tutorial was written specifically for the employees of House of Oak & Sofas but will work for anyone who is trying to access their BlueHost email through the webmail access. You can access the webmail BEFORE your site goes live by using the numerical ip address.

Sign in http://www.your-domain.com/webmail/



Authentication Required

A username and password are being requested by http://66.147.244.156:2095. The site says: "WebMail"

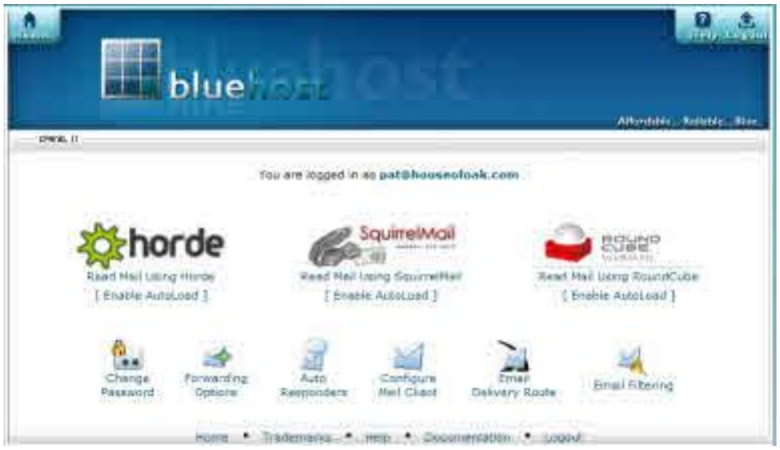
User Name: pat@houseofoak.com

Password: •••••••••

Figure 8 - Bluehost Webmail Signon.

Username: your full email address.
Password: your password.

Once signing in you have different email clients you can choose from to use Horde, SquirrelMail, or RoundCube. I recommend SquirrelMail.



You are logged in as pat@houseofoak.com

Figure 9 - BlueHost Email Programs.

Each of them can be set to autoload once you sign in.

Choose the one you find easiest to work with.

There is a spam filter with each account. Make sure you occasionally delete the items in that folder and empty the trash. I have found the spam filter to be pretty good in that no legitimate emails make it to the spam folder. If you find there are emails that are marked as spam that are not, let me know.

Tutorials on using various webmail clients

- http://tutorials.bluehost.com/webmail-squirrelmail/
- http://tutorials.bluehost.com/webmail-roundcube/
- http://tutorials.bluehost.com/webmail-horde/

Server Side Includes and BlueHost

NOTE: This tutorial assumes you do NOT have FrontPage Server Extensions installed.

I normally use Expression Web Design Time includes when working on websites that require repetitive content that will change from section to section like menus. The menus on this site are Design Time Includes. The only problem with using Design Time Includes on a site that does NOT have FrontPage Server Extensions installed is that you must republish ALL pages the include is part of. No big deal for a small site. But imagine a site with thousands of pages.

Perhaps you should consider Server Side Includes.

Does Your Server Support Server Side Includes?

First, you will need to check and see IF your server supports the use of Server Side Includes. It should be listed in the information on the type of package you have purchased. Two hosting services I have used are listed below.

- 1and1.com does support server side includes BUT only as part of their Business or Developer packages.
- BlueHost (affiliate link) offers only one hosting package and supports server side includes but by default is set up so that your pages need the shtml file extension to use them. But if your pages use .html or .htm, you will need to add the ability to handle ssi's within these file types.

So if this feature is important to you, check first to see if they are supported by the account you have chosen.

Setting Up BlueHost to allow SSI's in HTML Pages

By default, BlueHost is set up to allow you to use server side includes in pages with the file extension .shtml. But if your pages use the .html or .htm file extension, they will NOT work with this default setup. What to do? BlueHost has a tutorial on how to add the other file extensions so they will work. See Server Side Includes (SSI) The instructions advise:

Apache Handlers

This is what you would use to specify how to treat file types. By default, .html files will not run server side includes, but .shtml files will. To allow these you need to add a new handler for HTML:

To allow SSI, set the handler as 'server-parsed' and the extension as '.html' (without the quotes).

This will tell the server to read the .html files so that you may use these directives in them.

I signed in to my Control Panel > Advanced > Apache Handlers.

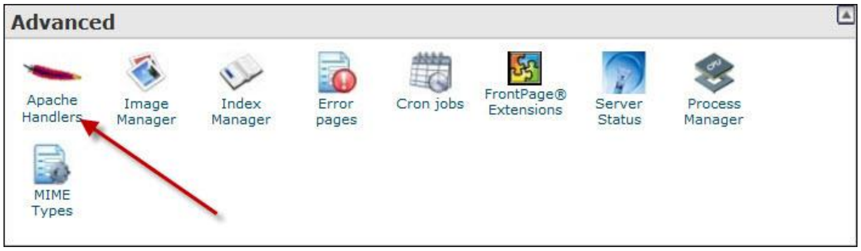


Figure 10 - Control Panel > Advanced > Apache Handlers.

Fill in the form as instructed:

Handler: server-parsed
Extension(s): .html .htm

I click **Add,** and check to make sure the site is still working. **Internal Server Error** is the message I see when I actually view ANY of my sites. Is part of the problem that I have pointed the main domain from the public_html folder to its own folder? Definitely!

WARNING: Do NOT use the Control Panel to create the apache handler if you have:

- redirected your main domain name from the public_html directory to its own folder.
- have any WordPress blogs set up.
- these are only the ones I am familiar with but I am sure there are others.

Editing the .htaccess file

Start Filezilla (or your ftp client)and sign in to your website. First, download the existing .htaccess file from your server. I save it to the actual website on my hard drive. If I am going to edit it, I save it as .htaccessold and work on the original copy. Make sure to set the server transfer type to ascii.

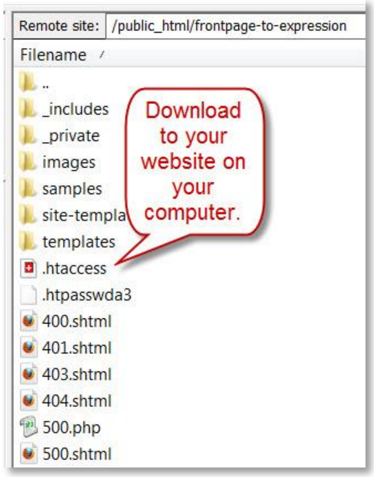


Figure 11 - Remote server showing .htaccess file.

NOTE: You will need to make sure the Filezilla is set to show the hidden files to view the .htaccess file. Click Server on the menu > Make sure Force showing hidden files is checked.

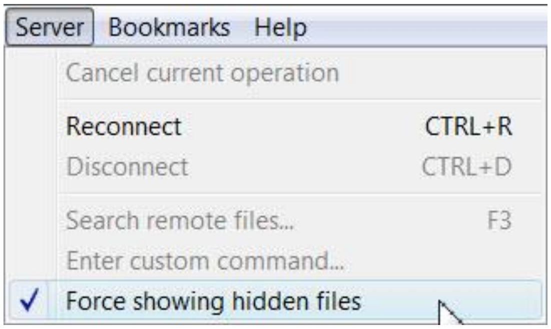


Figure 12 – Force FileZilla to show hidden files.

If you already have a .htaccess file, open the file in Notepad and add the following to the end of that file:

AddHandler server-parsed .html .htm

Make sure you do NOT leave off the dot at the beginning of the file extension.

Upload the edited file to your server replacing the one currently there. Check your site in a browser to make sure it still works. Remember to refresh so you are sure the site is working. Make sure and change the server transfer type back to Auto.

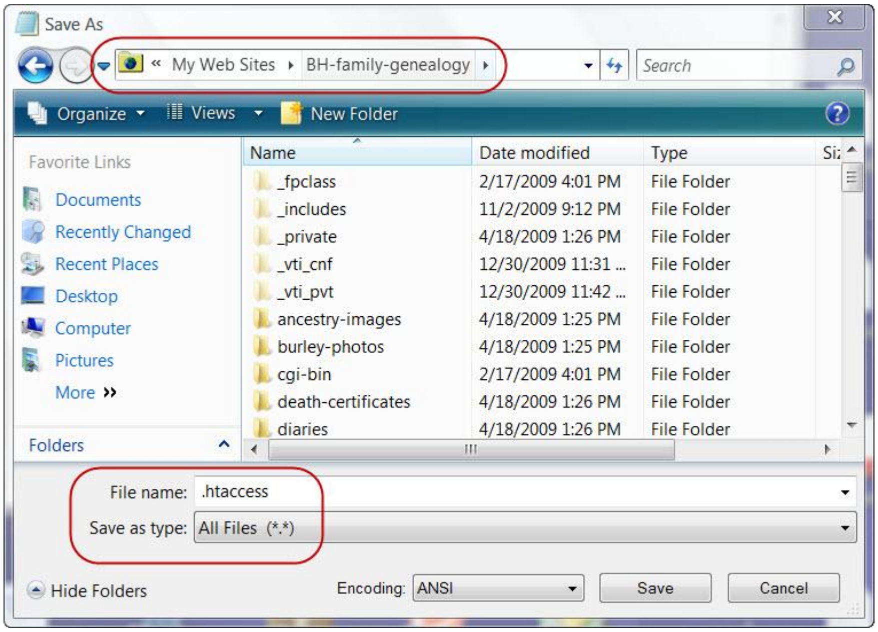
Creating a .htaccess file

If you do NOT currently have a .htaccess file on your server you can create one.

Start Notepad (or YOUR plain text editor) > From the Menu Click File > Save As

A dialog box will open. At the top of the box make sure it has the location of your website (on your computer) next to the Save in area.

To the right of File name type in .htaccess
To the right of Save as type, select All Files.
Click the save button.



Name	Date modified	Type
_fpclass	2/17/2009 4:01 PM	File Folder
_includes	11/2/2009 9:12 PM	File Folder
_private	4/18/2009 1:26 PM	File Folder
_vti_cnf	12/30/2009 11:31 ...	File Folder
_vti_pvt	12/30/2009 11:42 ...	File Folder
ancestry-images	4/18/2009 1:25 PM	File Folder
burley-photos	4/18/2009 1:25 PM	File Folder
cgi-bin	2/17/2009 4:01 PM	File Folder
death-certificates	4/18/2009 1:26 PM	File Folder
diaries	4/18/2009 1:26 PM	File Folder

File name: .htaccess
Save as type: All Files (*.*)
Encoding: ANSI

Figure 13 - Notepad Save As dialog box.

Add the following to that file and Save the file. **Do NOT forget the dot in front of the file extensions**.

AddHandler server-parsed .html .htm

Using Filezilla (or your ftp client) > Upload the file to your server.

You will need to add a new .htaccess file or the edit the current one for each domain you wish to use server side includes with. If you then check your site in your browser, and you see the message Internal Server Error, go back and double check to make sure you have a dot in front of the file extensions.

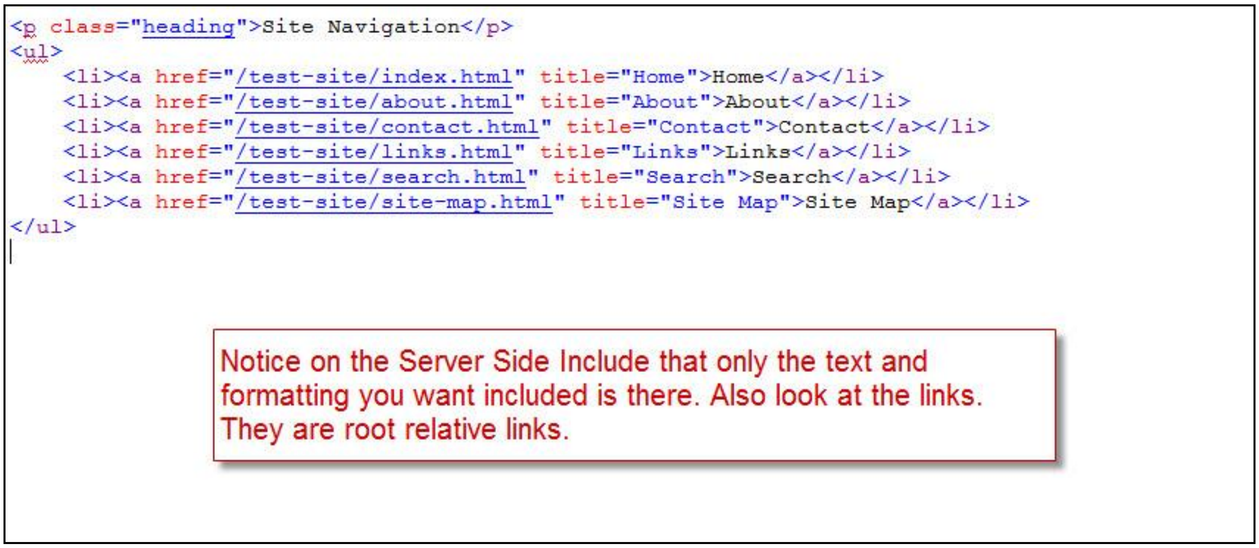
Create the include file

When you use a SSI file, the server takes the **entire content of the file and inserts it into the page**, replacing the include code. For example, the left side menu of this site changes from the main pages to the individual tutorial pages. If I use a SSI for the left side menu, I can make the change to just that file, save it and upload the include file to the server.

If you have a large web site using server side includes could save you considerable time and make the site easier to maintain. You need only make the change to the include file, save it and upload the include file to the server.

Create a file which contains the reusable code with the extension .htm or .html. You could also use .php BUT this tutorial does not cover those files.

If the ssi file is a menu with links to the various pages, the links can be either absolute (the full url) or root relative links (a URL relative to the document being served.

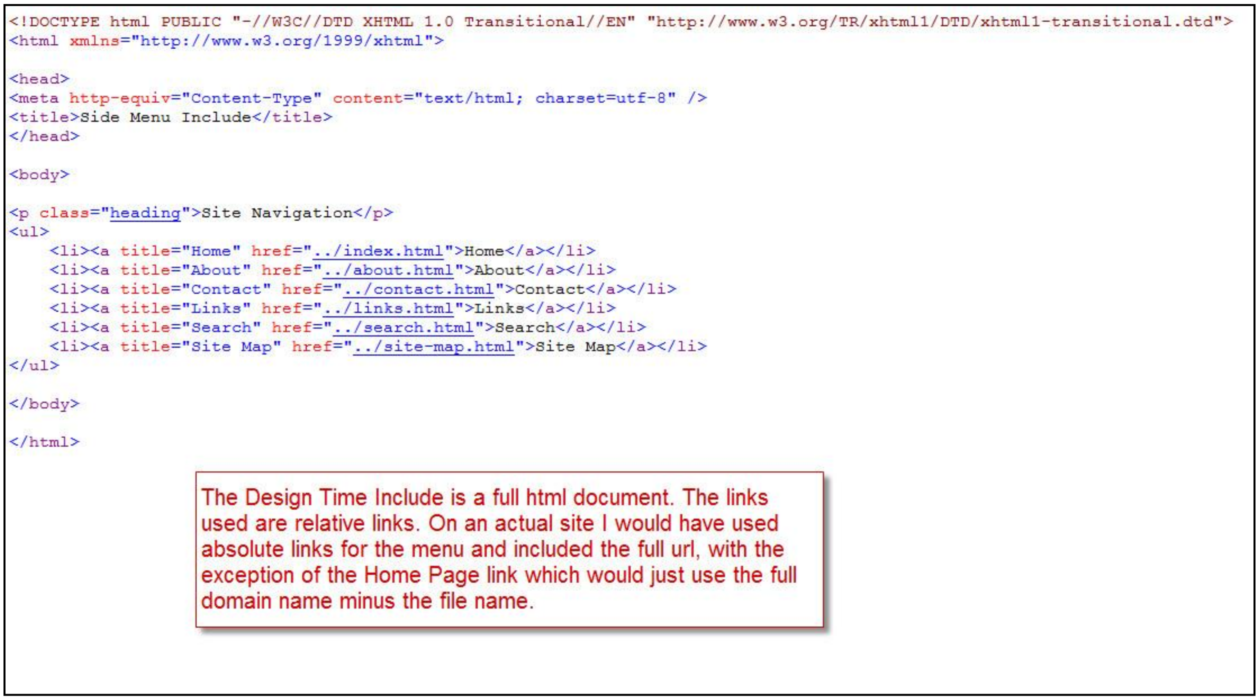


```
<p class="heading">Site Navigation</p>
<ul>
    <li><a href="/test-site/index.html" title="Home">Home</a></li>
    <li><a href="/test-site/about.html" title="About">About</a></li>
    <li><a href="/test-site/contact.html" title="Contact">Contact</a></li>
    <li><a href="/test-site/links.html" title="Links">Links</a></li>
    <li><a href="/test-site/search.html" title="Search">Search</a></li>
    <li><a href="/test-site/site-map.html" title="Site Map">Site Map</a></li>
</ul>
```

Notice on the Server Side Include that only the text and formatting you want included is there. Also look at the links. They are root relative links.

Figure 14 - Content of SSI.

NOTE FOR EXPRESSION WEB USERS: Unlike Server Side Includes, Design Time Includes need to be a full html document with at least the opening and closing <html> and <body> tags. The content between the opening <body>and closing </body> tags is what is inserted.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Side Menu Include</title>
</head>

<body>

<p class="heading">Site Navigation</p>
<ul>
    <li><a title="Home" href="../index.html">Home</a></li>
    <li><a title="About" href="../about.html">About</a></li>
    <li><a title="Contact" href="../contact.html">Contact</a></li>
    <li><a title="Links" href="../links.html">Links</a></li>
    <li><a title="Search" href="../search.html">Search</a></li>
    <li><a title="Site Map" href="../site-map.html">Site Map</a></li>
</ul>

</body>

</html>
```

The Design Time Include is a full html document. The links used are relative links. On an actual site I would have used absolute links for the menu and included the full url, with the exception of the Home Page link which would just use the full domain name minus the file name.

Figure 15 - Content of design time inlcude.

Server Side Includes into your page

The filename is given using a relative path and the referenced file which must reside on the same server as the calling page. The file extension of the included file doesn't matter. It can be .txt or .html or .htm, or .inc or even .ssi.

Go to the page that will use the include, and add the following code in the appropriate place:

<!--#include virtual="/directory/file-name.html" -->

The above is used if the include file is NOT in the same directory as the calling page. It should specify a URL relative to the document being served.

<!--#include file="file-name.html" -->

The above is used if the include file is in the same directory as the calling page. The file attribute is a file path, relative to the current directory.

If you use the wrong the syntax for the SSI or ask for a file that does not exist, you will see the error message below in place of the include.

[an error occurred while processing this directive]

Once the ssi is placed into the parent page you will not be able to tell it is an include. The code will look like it is part of the page.

Example Page - the left side menu is the include.

Using SSI's with Expression Web DWT's

If you are using Design Time Includes as part of your Dynamic Web Template and attach it to a page, any relative file paths are adjusted so they will continue to work no matter which directory you save the page too. This is NOT TRUE if you are using server side includes as part of your DWT. If you are going to use server side includes within your DWT, you will need to use root relative file paths for them to work no matter which directory level you use to save the page. Absolute file paths to the include file will fail since SSI will not work across domains.

I am going to use a test site I created within this domain (named test-site) to explain the file path you need to use within the DWT.

<!--#include virtual="/test-site/_includes/left-menu.html" -->

This assumes the starting at the root of your web /, then within a folder named test-site, there is a folder named _includes with the file named footer.html

If the file you are working with is in a folder called _includes located in the root of the web, the path to the include file becomes:

<!--#include virtual="/_includes/footer.html" -->

This assumes starting at the root of your web /, then within a folder named _includes and the filename footer.html

REMEMBER it is the actual path to the include file you are inserting into the DWT.

If you are using Design Time Includes within your pages, when you preview in your browser you will see the content of the include as part of your page. It often becomes a problem when you decide to use Server Side Include (SSI) for adding navigation menus etc. to your pages. Without the use of a Virtual Server on your own computer, you cannot "see" what your navigation (or other ssi's) actually looks like when checking through your pages.

NOTE: I would like to thank Barry Carlson of the FreePages list for coming up with this solution.

<!--#include virtual="/test-site/_includes/left-menu.html" -->
<script type="text/javascript" src="_includes/left-menu.js"></script>

To make this possible, open your file e.g. "left-menu.html " in Expression Web, highlight and copy into:

HTML to JavaScript Convertor

Click the 'Convert to Javascript' button, then Copy & Paste the result into a new text or javascript page and save as e.g. left-menu.js into the same Folder/Directory as 'left-menu.html. Before saving, edit out anything before <!-- and after //--> .

Just add the javascript link as in the example above to each of your pages, and lo and behold, you will have your server side includes showing locally.

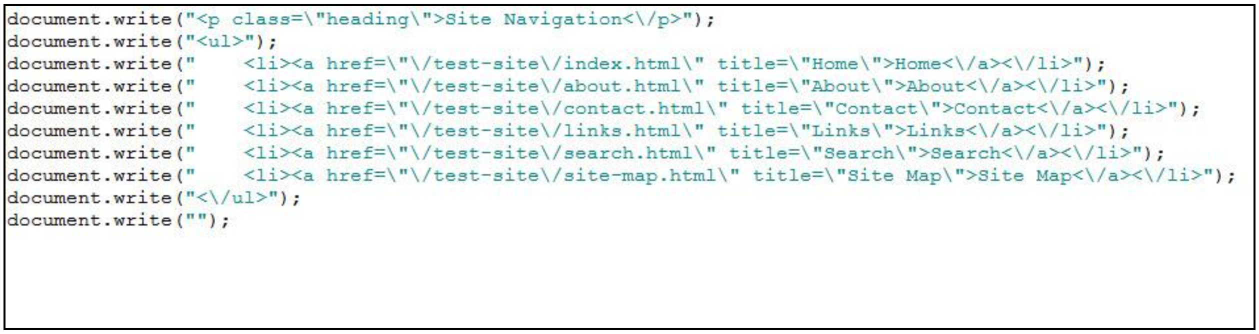
Mark the javascript file to exclude from publishing, and all the server will do is serve the

<!--#include virtual="/test-site/_includes/left-menu.html" -->

The javascript file is not there and no one will know any difference (if it's there, you'll get two of the include wherever it is!).

Example Page using the javascript described above.

The Server Side Include i view source to see the actual code.



```
document.write("<p class=\"heading\">Site Navigation<\/p>");
document.write("<ul>");
document.write("    <li><a href=\"\/test-site\/index.html\" title=\"Home\">Home<\/a><\/li>");
document.write("    <li><a href=\"\/test-site\/about.html\" title=\"About\">About<\/a><\/li>");
document.write("    <li><a href=\"\/test-site\/contact.html\" title=\"Contact\">Contact<\/a><\/li>");
document.write("    <li><a href=\"\/test-site\/links.html\" title=\"Links\">Links<\/a><\/li>");
document.write("    <li><a href=\"\/test-site\/search.html\" title=\"Search\">Search<\/a><\/li>");
document.write("    <li><a href=\"\/test-site\/site-map.html\" title=\"Site Map\">Site Map<\/a><\/li>");
document.write("<\/ul>");
document.write("");
```

Figure 16 - Javascript file since it is excluded from publishing.

While using Server Side Includes may make site maintenance a tiny bit easier in that you do not have to upload the parent page when updating the include page, I still prefer using the Design Time Include feature of Expression Web.

Additional Resources

- HTML Goodies
- Google's Search Results for Server Side Includes
- Lorrie Laskey's Server Side Includes
- Apache Tutorial: Introduction to Server Side Includes
- Server Side Includes by Wikipedia
- Using the .htaccess File from Wise-Women Tutorials

SSI Issues

Having the server parse documents is a double edged sword. It can be costly for heavily loaded servers to perform parsing of files while sending them. Further, it can be considered a security risk to have average users executing commands as the server's User. If you disable the exec option, this danger is mitigated, but the performance issue remains. You should consider these items carefully before activating server-side includes on your server.
~ http://hoohoo.ncsa.illinois.edu/docs/tutorials/includes.html ~

Table of Figures