CF Custom SEO URL Rewrite - Yet another way to do this

Posted At : November 18, 2011 9:21 AM | Posted By : Tim Garver
Related Categories: Projects

I know the title is a bit vague, and I know there 101 ways to tackle this and probably the most accepted way is at the web server level with MOD re-write or ISAPI. But at that level its harder to incorporate business logic or custom data.

So what if you are on a shared host that does not allow MOD-REWRITE or the IIS version ISAPI rewrite, but will allow a custom 404 file, AND the SEO requirement is to hide the real file name?

For instance, suppose you need the URL to look something like this:

http://www.server.com/{1st resource}/{2nd resource}/{3rd resource}

Maybe a search URL like:

http://www.server.com/bmw/4 door/blue/

OR

http://www.server.com/Charlotte/2 story/4 bath rooms/

So the CGI var will end up being something like this depending on your web server:

```
404;http://server.com:80/{1}/{2}/{3}
```

CGI.QUERY_STRING from IIS6 Where {1} = the first node in the URL, I am using the term node there may be a better term to use. But its the first "folder" in the URL string.

In Apache, the query_String is sometimes empty and it puts the values you need in the http_referer cgi field.

To parse this I used this script, its not perfect and I am sure there is a regex for it but this is what i came up with:

```
// setup the url string, parse the cgi struct to pull out all past host name
if(cgi.server_software NEQ "Apache" AND len(cgi.query_string)){
  qString = ListDeleteAt(MID(ListLast(cgi.QUERY_STRING,";"),LEN(ListFirst(ListLast(cgi.QUERY_STRING,";"),":")) + 4,150),1,"/");
} else if(cgi.server_software EQ "Apache" AND len(cgi.query_string)) {
  qString = cgi.QUERY_STRING;
} else if(len(cgi.http_referer)){
  qString = ListDeleteAt(cgi.http_referer,1,"/");
  qString = ListDeleteAt(qString,1,"/");
} else {
  qString = "";
}

node.thisDirectory = GetDirectoryFromPath(ExpandPath("*.*"));
node.rootisDir = directoryExists(node.thisDirectory & listFirst(qString,"/"));
node.lastNode = ListLast(qString,"/");  //last url node varialbe
node.firstNode = ListFirst(qString,"/");  //first url node var
node.nodeLen = ListLen(qString,"/");  //number of url nodes
node.nodeArray = listToArray(ListFirst(qString,"!"),"/");  // node array
```

for instance here is a dump of the functions return data for this URL: http://server/profile/tim

In my mappings I have an ID attribute setup for this map. When I loop over the maps, I build a return data attribute with the map ID value as the name of the data attribute and the value of the last or second node as the value of the data attribute.

```
mapping.profile = structNew();
mapping.profile.file = "profile.cfm";   //file that gets included
mapping.profile.id ="userid";  //variable to assign the numeric value if one is found in the url path
mapping.profile.title = "My Profile";
mapping.profile.secured = true;         //mapping is secured
mapping.profile.role = "user";          //secured by role
mapping.profile.path = arrayNew(1);
mapping.profile.path[1] = "profile";  //SEO url path
mapping.profile.path[2] = "me";       //SEO url path

// the rest of the code creates the other mappings and then just loop over the mappings to match the path array values to the node array values. if a match is found include the mapping file, otherwise include the missing
```

One problem I found was that when a hash sign was in the URL it was not represented correctly by the query string from the web server. IIS just ignores it all together, So I added the ability to replace a # sign in a url with a ! sign. Here is the URL for this dump: http://server/about!tim

Notice the anchor value is tim. On the actual pages, we just check for the anchor to see if there is a value we used jQuery to jump to that anchor in the page.

These dump's don't show all the values, just a few to demonstrate most of the values.

So there you have it, if you want to give it a shot, jump on over to my RIAForge.org project

I would love to hear from anyone else who has had to do this for an odd reason and could not use the standard rewrite tools that are out there today.