



## ColdFusion Site Skinning Tutorial

---

Are you always looking for something new and fun to offer your website visitors? If so, perhaps you'd like to offer them the ability to choose the site design they'd prefer to use whilst browsing your website. Hopefully this tutorial will allow you to do just that - site skinning. You should have a basic understanding of ColdFusion and Cascading Style Sheets (CSS) to use this system.

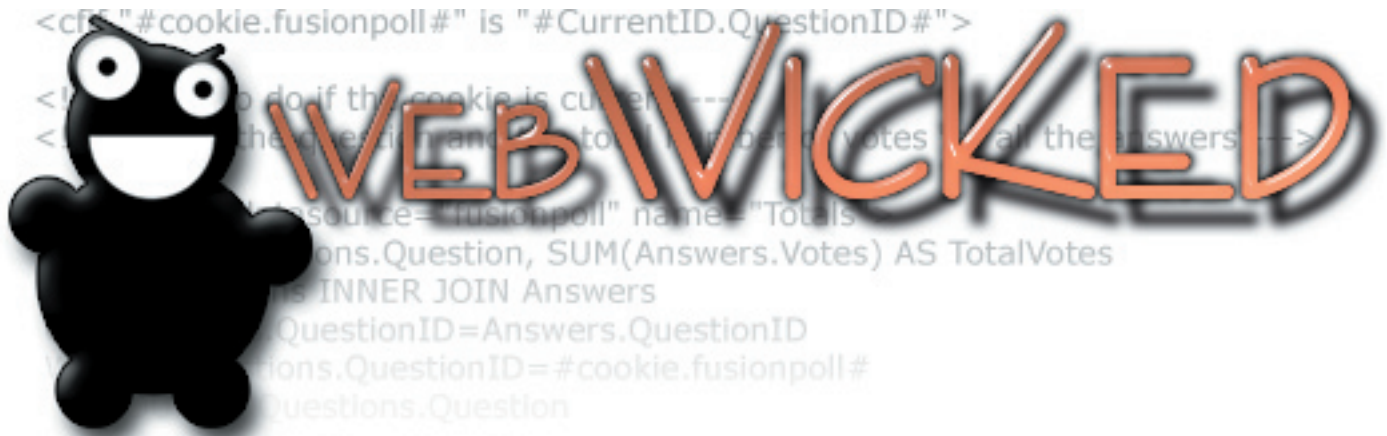
The pages we need to create are the application.cfm, change.cfm, header.cfm, footer.cfm and index.cfm pages. If this seems like a lot of pages to you, don't worry. They're all small pages :)

The basic concept behind this skinning system is that when the form is used, a choice is made that will select which CSS to use when browsing the site. So our CSS must contain all of the website textual and graphical content in it. Let's look at how I set out the CSSs for this example.

I broke each table on my website into 9 sections. Top left, top middle, top right, middle left, middle (this is where the content goes for each table), middle right, bottom left, bottom middle and bottom right. you could simply set colours or images for these elements. I chose images so that each design could be totally different. So in the CSS I had to create a class element for each section of the table:

```
td.topleft {  
background-image: url(site01/images/design_01.gif);  
font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;  
font-size: 12px;  
background-repeat: no-repeat;  
}
```

```
td.middle {  
font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;  
font-size: 12px;  
background-color: #9B9B93;  
padding-bottom: 5px;  
padding-left: 5px;  
padding-right: 5px;  
padding-top: 5px;  
}
```



## ColdFusion Site Skinning Tutorial

---

You can see that the topleft part of the CSS has the image details whilst the middle section has padding for the text that will appear in that part of the table as well as the text details. There is a similar piece of code for each different element of the table in the CSS. Each different site skin will have different CSS values and its own actual CSS. The different design elements you can add to your website are up to you and your knowledge of CSSs. Now onto the ColdFusion.

### application.cfm

It is on this page that we allow for the setting of our cookie and where we then check for any pre-existing cookie.

```
<CFAPPLICATION NAME="Cookies"
SessionManagement="Yes"
setDomainCookies="Yes"
setClientCookies="Yes">
```

```
<cfif isdefined ("cookie.siteskin")>
<cfif "#cookie.siteskin#" is "green">
<link href="design01.css" rel="stylesheet" type="text/css">
<cfelseif "#cookie.siteskin#" is "maroon">
<link href="design02.css" rel="stylesheet" type="text/css">
<cfelse>
<link href="design03.css" rel="stylesheet" type="text/css">
</cfif>
<cfelse>
<link href="design01.css" rel="stylesheet" type="text/css">
</cfif>
```

What the code above is doing is looking to see if the cookie existed. If it did, then it checks to see what the value of the cookie was and links to the related CSS. for example, if the cookie value was "maroon", the [design02.css](#) is set. If the cookie did not exist, the [design01.css](#) is set (our default design). From here we move on to the index.cfm page.



## ColdFusion Site Skinning Tutorial

---

### index.cfm

This is the page where we will offer the choice of site skins to the website visitor. This page (as well as all others in your site) should use the header and footer pages as inclusions just so it is easier for you to maintain your website later. If you decide to add or subtract a site skin later, it is easier to just edit the header and footer (as well as the application.cfm page) instead of every page in your website.

The form itself looks simple enough, purely because it is simple :)

```
<form name="siteskin" method="post" action="change.cfm">
<select name="siteskins">
<option value="green">Green</option>
<option value="maroon">Maroon</option>
<option value="blue">Blue</option>
</select>
<input type="submit" name="Submit" value="Submit">
</form>
```

The form offers a list of choices (one for each site skin on offer) and the submit button. On pressing the submit button, the form options are sent to the change.cfm page which we'll cover later. Before we move on to the change.cfm page, we should have a closer look at the header and footer pages.

### header.cfm

The header page works in the same fashion as the application.cfm page works. It checks for the existence of a cookie and displays the corresponding header for that cookie.

```
<cfif isdefined ("cookie.siteskin")>
<cfif "#cookie.siteskin#" is "green">
<cfinclude template="site01/head01.cfm">
<cfelseif "#cookie.siteskin#" is "maroon">
<cfinclude template="site02/head02.cfm">
<cfelse>
<cfinclude template="site03/head03.cfm">
</cfif>
<cfelse>
<cfinclude template="site01/head01.cfm">
</cfif>
```



## ColdFusion Site Skinning Tutorial

---

As you can see, I setup each site skin to use a separate subdirectory on my site. This was again to help keep each site skin separate from the other and easier to maintain down the track if need be. If no cookie is in existence, the default header ([site01/head01.cfm](#)) is used.

### footer.cfm

The footer page works the same as the header page with the footer being matched to the corresponding cookie value.

```
<cfif isdefined ("cookie.siteskin")>
<cfif "#cookie.siteskin#" is "green">
<cfinclude template="site01/foot01.cfm">
<cfelseif "#cookie.siteskin#" is "maroon">
<cfinclude template="site02/foot02.cfm">
<cfelse>
<cfinclude template="site03/foot03.cfm">
</cfif>
<cfelse>
<cfinclude template="site01/foot01.cfm">
</cfif>
```

### change.cfm

The change.cfm page sets the cookie value depending on the choice made by the form user. As you are hopefully aware, you can not set a cookie in ColdFusion and then use the `cflocation` tag to send your website visitor to another page. Trying to do so will cause the cookie to not be set.

```
<cfif isDefined("form.siteskins")>
<cfcookie name="siteskin" value="#form.siteskins#" expires="never">
<meta http-equiv="refresh" content="0;URL=index.cfm">
</cfif>
```

So I used simple meta tags to redirect the user to the desired page and refresh the page so that the new cookie setting is forced to take effect. Otherwise, the chosen setting will not appear until the next page view.

So that is the basic system I created. This system is basic yet hopefully good enough for you to use on your own website with ease. Try it out and have fun adding site skins to your own website.