

Topic Notes: Server Side Includes

We will take a brief look at a tool that might be useful to some groups for your design projects.

When developing a site with more than a handful of pages, we will often try to accomplish a consistent look and feel for all of the pages in the site. Our course web page is a simple example: each page has the same header across the top. We have already seen one technique that helps here: linking to the same stylesheet from each page:

```
<link rel=stylesheet
      href="http://courses.teresco.org/commonstyle.css"
      type="text/css" />
```

As we saw in one of the labs, making changes to this CSS will make changes to all pages on the site that link to this stylesheet.

But we still need to get that HTML text into each file that defines the course number and name, and provides the text and links for the menu bar. In the case of our course web pages, the system used to develop all of the course materials adds this in automatically to each page. But what if you are developing pages right in HTML?

We could accomplish this using JavaScript. Each file in the site links to the same JavaScript functions and calls one of those functions while the page is first being rendered.

See Example: [page1js.html](#) See Example: [page2js.html](#)

However, this relies on the client to execute the JavaScript code (not all clients do).

Another mechanism to do this is called a *server side include (SSI)*. These are directives placed within HTML documents that direct the web *server* (i.e., not the browser/client) to insert some information in the document before sending it to the client for rendering.

The availability of this capability depends on the web server and its configuration. We will look at the mechanism as available in the Apache httpd server as configured on the [teresco.org](#) and [sienacs.com](#) web servers.

The idea is straightforward: we place a specific directive within our HTML document in the place where we want the common information to be placed. Then we put that information into another file that gets “included” by the others.

We consider an example. We have pages in files `page2.shtml`:

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

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>A Page with an SSI</title>
</head>

<body>
<p>This part of the document is only on Page 1.</p>
<!--#include virtual="common.html" -->
</body>
</html>

```

and in page2.shtml:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Another Page with an SSI</title>
</head>
<body>
<p>This part of the document is only on Page 2.</p>
<!--#include virtual="common.html" -->
</body>
</html>

```

and another file common.html:

```

<p>This is included from <tt>common.html</tt> and is visible in both
page 1 and page 2. Any change made in <tt>common.html</tt> will be
seen in both page 1 and page 2.</p>

```

If we install these files onto a web server with SSI enabled, page1.shtml will be served to a client as:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>A Page with an SSI</title>
</head>

<body>
<p>This part of the document is only on Page 1.</p>
<p>This is included from <tt>common.html</tt> and is visible in both
page 1 and page 2. Any change made in <tt>common.html</tt> will be

```

```
    seen in both page 1 and page 2.</p>  
</body>  
</html>
```

And similarly for `page2.shtml`.

How does this work?

First, notice that the two files we wish to be able to display have an extension of `.shtml` instead of the usual `.html`. This instructs the web server to look for server side include directives and apply them before sending the file to a client.

The `include` directive itself looks like an HTML comment, but it is in a special format that will be interpreted by the web server. The client will never see the directly. It will be replaced with the contents of the specified file from the web server.

The included file, called `common.html` here, can be anything, but it is important that it makes sense in the context of the document that includes it. For example, it should not have a `DOCTYPE` or `<html>` element or any similar structure.

Another example: the `http://j.teresco.org` site.

Some web servers do not support SSI at all, some require the `.shtml` extension on files that should be processed using SSI, and others will recognize SSI on all files.

There is more we can do with SSI, but these simple includes are the most common.