

Advanced Manila Coding Using Macros

Introduction

At this point you have created the layout that you wish to convert into a manila theme. It really doesn't matter what software you use to create the html as long as it is generally standard html. In the experience of the workshop facilitators, Microsoft Front Page does not tend to be a good editor for manila themes as it typically inserts Microsoft-specific tags that do not serve well from a Frontier server.

The primary thing to keep in mind while creating this is that within the education environment you typically have no idea what someone may put into the page using your theme. One person may use nothing but text to post homework assignments and notes. The instructor in the room next door may wish to post field trip photos, PDF documents and Quicktime VR movies.

Manila Tags

Manila tags are straightforward. They are simple tags that say exactly what they are. The more common ones are:

- `{bodytext}`* This is the big tag. The `{bodytext}` tag defines where the body of the document will reside. Even more so than the `{calendar}` tag, this one requires a lot of flexibility. A user may only post a list of short words that may take up all of 200 pixels in width. Another user may post 640 pixel wide photographs that they took with a digital camera. Your theme needs to allow for all of these without any of them breaking your design. Table and cell widths based on percentages rather than pixel widths may give you more play room here.
- `{calendar}`* In Manila, a calendar is useful when a user is managing their website with news stories. The calendar allows the site visitor to navigate to stories that were published on past days. Outside of the body text area, this is probably your biggest design consideration (read: pain in the neck). The calendar can take up a considerable amount of your horizontal and vertical web page space. It needs to be placed such that the layout doesn't look awkward without it, or with it, as the user decides. The appearance of the calendar is set on the advanced preferences page.
- `{cascadingStyleSheet}`* Place this tag in the `<head>` section of your HTML document. Place the actual CSS code on the advanced preferences page.
- `{day}` Display the number of the current day.
- `{dayName}` Display the name of the current day.
- `{hierarchyPath}` Display the hierarchy path to the page.
- `{hostingSuite.createSiteForm ()}`
If you are not using the front page of your manila server as a "Create a Manila Site" page, you may want to include the create-a-site form on another page instead. Usage of this tag will place the create-a-site form on whatever page you wish.

<code>{hostingSuite.listSites (true)}</code>	If you would like to provide a way for visitors to search your server for a particular site, this is the tag to use. Manila will place a search form at the location of this macro that guests can use to search for the exact site that they are after.
<code>{includeMessage (22,flProcessMacros:true,flDGlink:false)}</code>	Use this tag to create alternate editable regions within a page. Manila always puts at least one editable region on each page. This region is defined by the <code>{bodytext}</code> tag. There are times when you may want to have other sections of your page editable in the same kind of way. The <code>{includeMessage...}</code> tag allows you to enter the contents of another story at the point of the macro. In the case of the tag given, the contents of story number 22 will be entered. Change the story number to the one appropriate for your site.
<code>{javascript}* </code>	Place this tag in the <code><head></code> section of your HTML document. Place the actual javascript code on the advanced preferences page.
<code>{lastUpdate}</code>	Display the date and time a page was last updated.
<code>{lastUpdateString}</code>	Display the date and time a page was last updated, using a friendly format.
<code>{membershipBox}* </code>	If you have membership enabled for your site, this is where manila places the membership box. When a user has not logged in to your site, or is a non-member guest, it can take that user to a page where they can either login or sign-up as a member. If a member is logged in, it gives that user easy access to logout.
<code>{meta}* </code>	Place this tag in the <code><head></code> section of your HTML document. These are non-editable tags that manila uses to aid web robots as they browse your site for search engine inclusion.
<code>{month}</code>	Display the number of the current month.
<code>{monthName}</code>	Display the name of the current month.
<code>{navigatorLinks}* </code>	This is where the in-site navigation will get placed. The items listed in the navigation can be edited via XML on the advanced preferences page. When you click on the "prefs" link in the editorial menu, the navigation links will be replaced with preference options. You can modify the appearance of the navigation links near the bottom of the advanced preferences page.
<code>{now}</code>	Display the current time and date on the server.
<code>{postTime}</code>	Display the time and date when a page was originally posted.
<code>{printFriendlyLink ("Print-friendly version of this page")}</code>	This is the tag to insert if you would like visitors to be able to print a printer-friendly copy of the page. Generally a printer-friendly version of pages contains no navigation or unnecessary graphics. Place within the quotes the text that you wish visitors to click on to access the printer-friendly version. You can edit the printer friendly template in the advanced preferences.
<code>{siteName}</code>	The site name as given by the site owner on the appearance preferences page. You will typically want to place this in a very prominent position near the top of the theme.

{tagLine}	The tag line as given by the site owner on the appearance preferences page. This is an optional tag, but one that users often find great use for. As an example, a user may have a site name of “Atlantis High Guidance Office”. An appropriate tag line may be, “Helping future atlantians prepare for the future”.
{themeLink}	This will get placed automatically at the bottom of pages using your theme. It will link them back to the home page of the theme creator. When users apply your theme to their site, this tag is automatically inserted at the bottom of each page.
{whoAmI}	Display the name of the member looking at the page. Returns nothing if empty.
{year}	Display the number of the current year.

*required tags noted by **

Placement and Theme Design Considerations

Some of the tags place content that is generally consistent—for example, the calendar always looks the same. The trick with the calendar is that it is an optional page element. When it is used, it is a rather prominent part of the design, however your layout can’t look like something major is missing when the calendar is absent. Conversely, there are elements of your theme such as the title that always display but have a sometimes unpredictable size. One user may simply enter their name as the title. Another user may enter something with a bit more length such as “Neptune’s Sixth-Grade Deep Sea Wanderings”. Again, your theme design needs to be flexible enough to accommodate anything, within reason, without looking “off”.

Modifying Your HTML for Use in Manila

Begin by either copying and pasting the html code from your graphical editor into your text editor, or by opening the file itself.

Editing the <head> Section

The head section of your document will appear similar to the following:

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
p { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
font-weight: normal; color: #000000}
h1 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
18px; font-weight: bold; color: #006699}
h2 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
14px; font-weight: bold; color: #006699}
-->
</style>
</head>
```

We need to modify this code to tell manila where to place the title, css, javascript and meta tags that we have created. First, copy the entire <style> section beginning with the <style> tag and ending with the </style> tag. Create a new document in your text editor and paste the cascading style sheet code into there. Return to main code. In the place of the cascading style sheets, place a {cascadingStyleSheets} manila tag. Replace any meta tags that your graphical editor may have placed in your code with a single {meta} manila tag. Even though we did not do any javascript in this lesson, manila needs to know where this would fall, so also place the {javaScript} tag in the <head> section. With the above changes, your <head> will look like this:

```

<html>
<head>
<title>{title}</title>
{meta}
{javascript}
{cascadingStyleSheet}
</head>

```

That wasn't so bad. It doesn't really get any more difficult than that. Actually, it gets easier. When creating our theme using Dreamweaver in the previous lesson, we left textual clues for ourselves to aid us in the next section.

Editing the <body> Section

Let's look at the <body> section of our code:

```

<body bgcolor="#3399CC" text="#000000" background="bg.gif">
<table width="600" border="0" cellspacing="0" cellpadding="0"
align="center">
<tr bgcolor="#006699">
<td rowspan="5"></td>
<td></td>
<td rowspan="5"></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td bgcolor="#006699"></td>
</tr>
<tr>
<td>
<table width="100%" border="0" cellspacing="0" cellpadding="10">
<tr align="left" valign="top">
<td width="100" bgcolor="EFEFDB">
<p>calendar</p>
<p>navigation</p>
<p>membership</p>
</td>
<td width="440" bgcolor="#FFFFFF">
<p>editorial menu</p>
<p>page title</p>
<p>body text</p>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td bgcolor="#006699"></td>
</tr>
</table>
</body>

```

The first thing that we need to do is replace our placeholder text with manila tags. Do this as shown:

```

{calendar}
{navigatorLinks}<br>
{membershipBox}
</td>
<td width="440" bgcolor="#FFFFFF">
<p>{page title}<br>
{editorialMenu}</p>
<p>{body text}</p>

```

Second, we need to fix our image tags. Currently they assume that the graphics are in the same directory as the page itself. That will not be the case for a Manila theme. You may have a separate server that your graphics will be served from. Perhaps you are serving the graphics from your Frontier server. Either way, you need to define your graphics with absolute URLs.

We will make the following changes to all instances of the affected graphics:

```



```

Editing Cascading Style Sheets

Now we need to make a small change to our css code. Return to the text editor window where you have that stored. It should look similar to the following:

```
<style type="text/css">
p { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
font-weight: normal; color: #000000}
h1 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
18px; font-weight: bold; color: #006699}
h2 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
16px; font-weight: bold; color: #006699}
</style>
```

The change that we need to make involves the <p> tag that we defined. We want all body text to appear with the properties that we defined for <p>. Body text doesn't always fall inside of <p> tags, so we need to broaden the scope of the tag to include other possible tags that body text may fall in. We will add the <body> tag and the <td> tag as ways to catch the rest of our text.

We also need to add a third style sheet definition for <h3>. This will be the same as <h1> except that the text it define will be white, as to show more clearly on a dark background.

```
<style type="text/css">
p,body,td { font-family: Verdana, Arial, Helvetica, sans-serif; font-
size: 10px; font-weight: normal; color: #000000}
h1 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
18px; font-weight: bold; color: #006699}
h2 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
16px; font-weight: bold; color: #006699}
h3 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
18px; font-weight: bold; color: #FFFFFF}
</style>
```

If you haven't worked directly with css code before, you should be able to see here that if and when you decide to make changes, it is not a difficult thing to figure out. If you decide that your text is too small or a headline too big, simply come directly to the code and enter new values.

Establishing the Theme in Manila

With the above changes made, our HTML should be ready for Manila.

Upload Your Graphics

First, ensure that all graphics that are used by the theme have already been uploaded to the location that you specified when rewriting the image tags. As soon as we enter the HTML text into Manila, it will need access to them.

Placing HTML into Manila

Keeping your text editor open, launch a web browser and login to your Manila site with your given email address and password. Once logged in, you will find the editorial menu near the top of the page.

Click on the “Prefs” link. You will be taken to the front page of the preferences section. Find the “Advanced” link at the end of the preferences menu and select it. You are now ready to make changes to the advanced preferences of your Manila account.

Return to your text editor and bring the window containing your HTML code to the front. Select the entire HTML text and copy it. Go back to your browser and find the template section. You will notice that there are actually two templates areas. The first is the template that will be applied to your entire site. The second is an optional template that you can have applied to only your front page, if you wish for it to have a different appearance than the rest of your site. There is check box following the home page template to turn it on or off.

Highlight everything in the template section and delete it. Now, paste the contents of your clipboard, your HTML, into the templates area. Hit “Submit”. You should notice an instant change in the appearance of your page.

Placing Cascading Style Sheets into Manila

There are just a couple of things left to do on the advanced preferences page. Find the Cascading Style Sheets area, immediately following the templates. Return to your text editor and find the window containing the css code that we worked on earlier. Copy it and bring your web browser back to the front. Delete any code that is currently sitting in the cascading style sheets area and paste in your own code. Hit the submit button. Again, you should notice instant changes as your text takes on the properties defined in your new style sheets.

Clean Up Navigation Appearance

Scroll down the page and find the section titled “Navigation Appearance”. We want the navigation to appear down the left side of our page. To put a single line break after each menu item, the separator format must be set to “
”. If we had a menu that appeared across the top of our page we could do something like “ , ” or “ | ”. Hit “submit” again to lock in your changes.

Fine Tuning the Theme

That should hold us for a while on the advanced preferences. Let’s move to some other preference panels to clean up the appearance of our theme. In your preferences menu, click on the “Appearance” link near the beginning of the menu.

Appearance: Site Name and Tag Line

The first two items in the appearance preferences panel allow you to define the site name and tag line. We’ve already defined where these will fall within the theme with the {siteName} and {tagLine} manila tags. You can place values into these if you wish and they will be applied to your site. These values do not become a part of your theme. Each user is free to redefine these as they wish.

Appearance: Colors

The most important part of the appearance preferences panel, as it relates to theme building, is the colors section. The color defined here will become part of the theme and define how many things appear within pages utilizing your theme. Due to the nature of the theme that we are building today, some of these can be ignored. We have defined a background image, therefore the background color is irrelevant. We have defined a body text color within our cascading style sheets and thus have no need for the text color setting allowed here. We have not defined the “Visited Link” and “Active Link” colors in our style sheet. Delete any value that may be sitting here and leave the field blank to allow the end-user’s browser to use its own color settings. We will be turning membership off for today, so the “Members box” settings do not need to be modified. We definitely want to redefine the “Separators” color to match our theme. Separators are those lines that fall between sections of pages. An example of a separator would be the line on the appearance preferences page that says “Colors:”. Change this color to “006699”. The default color for “Separator text” is “White”. Since we have defined a dark color for the separators, white will work just fine. We will not be using discussion groups, but will from time to time

encounter an admin box. Change the colors for “Admin Box Header” to “006699”, Admin Box Header Text” to “White”, “Admin Box Background” to “EEEEEO” and “Admin Box Text” to “Black”.

Appearance: Home Page

This section mainly applies to the use of news items on the home page. We will be discussing their use later and will be returning to this section of the preferences to make changes. For now, make sure that the “Do you want a calendar on your home page?” and “Do you want an editing form on the home page?” values are both set to “no”.

That wraps it for the appearance page. Click “Submit” at the bottom of the page and let’s move on.

Membership

Manila has a lot of features built into it. Some of those features include the abilities to run discussion lists and send html formatted emails to a membership list. As we are building a classroom page, we are not interested in allowing others to join our page. In order to prevent others from signing on as members of our site, we will hide the membership box that currently appears on our site.

Click on the “Membership” link in the preferences menu. The first option is “Do you want membership enabled for this site?”. Set this to “No”. With the first option set to “No”, the rest of the page become irrelevant. Move to the bottom of the page and hit “Submit”.

Saving Your Settings as a Theme

Once you have your site looking the way that you want it to, it is time to actually save it as a theme. Saving your settings as a theme will allow you to publish this appearance for others to use on their own sites.

Neither saving as a theme nor using a theme, locks you into an unchangeable format. Themes are simply starting places. Some users will apply a theme and never move beyond the appearance of the theme. Others will apply the theme and then make modifications, depending on their skill level, that better suit their needs.

Return to the advanced preferences page by clicking on the word “Advanced” in the preferences menu. Scroll all the way to the bottom of the page to find the “Theme” section. Change the setting “Do you want to make this site cloneable as a theme?” to “Yes”. Give your theme a title such as “Atlantis HS Staff Pages”. Enter your own name and email address for the “Author:” and “Author email address:” settings. Finally enter a description of your theme and hit “Submit”.

Downloading the Theme

Once Manila has built your theme, it will be downloadable from the “Theme” section of the advanced preferences page. Where the text reads, “Your theme can be downloaded [here](#).”, click on the “here” link and your theme will be downloaded as a FFTB file to your computer.

Publishing Your Theme

Anybody can create a theme and download it as a file. Only a Frontier system administrator can load the theme to the Manila server so that others can actually use it.

To load the file on the server, you will need to have file access of one sort or another to that box.

Find the folder that Frontier was loaded into. Within that folder open the *Guest Databases > Ops > Themes* directory. You should find a few themes already loaded here. Place the FFTB file that you downloaded in this directory.

Accessing Your Theme

Go back to your web browser. In order for your current site to recognize a change in the themes directory, you will need to logout. Click on the “logoff” link in the editorial menu. A screen will appear to confirm your decision. Complete the logout.

Type in the url to your site and return to the front page. There is a link in the body text on the front page to login. Follow that link, insert your given email address and password, and login.

The themes settings are found in the preferences. Click on the “prefs” link in the editorial menu. You will find “Themes” in the preferences menu. Be careful that you do not select “Discussion Group Themes”. A list of the currently available themes will be presented to you. The new theme should be included as a member of this list. While it won’t make any changes to your site, go ahead and select the new theme. Click “Submit”. Manila will ask you to confirm your decision. Say “Yes.”

You have now created a Manila theme. We began the previous lesson talking about the need to separate form from content. With the form now complete, we can move forward with creating and placing content.