



Rako Studios » Media » Suffering-with-software » Creating back issues in the Archive

Creating back issues in the Archive

**Fallacies: 1) content and presentation are separate,
2) publishing on the web is not like a paper magazine.**



Even the power-mongering programmers that want to take over from the art and marketing departments have to admit that well, maybe size information should be attached to images. If you want to see how lame programmers' presentation skills are just compare any page in a paper magazine to the corresponding web page.

Some of this is due to the fact that html is a markup language and not a page-definition language. So to display a decent page on a computer, people use pdf files, an evil printer-driver spin-off. That just drives people from our sites and allows them to email the pages to each other oblivious to the web community we try to build.

Walk into any Barnes & Noble bookstore. Look at the rack of magazines. Each has a bright enticing cover with a big bold design. If you want to see what these magazines' web pages look like, just open them all up to the table of contents page and put them back on the rack.

Geek Aspergers anti-social programmers can try to deny 200 years of visual articles and publishing expertise, but only at their own peril. Sooner or later the web teams will learn that that the reason magazine covers don't look like web pages is that art directors are much better at getting people to open the magazine and start browsing.

So an important feature of my web sites is that the home page is like a monthly magazine cover. I doubt I will be able to come up with a new home page every month, but I feel that it is good to have episodic issues of a website. One important implication of this is that each monthly homepage should have its own unique stylesheet (CSS). So that got me thinking how to implement this feature in a Movable Type website. I don't claim to have an optimum or even a good method, but it works.

What you do is create a new home page by using the Movable Type engine in the Archive blog you just set up. You don't use an entry page. That would be hopeless, since the entry page is just a way to put in some basic text and html, and you would have no real control of the home page. What you do is use a custom archive template. You clone the installed archive.html template and rename it to the year and month, such as 2011-09. For the output of this cloned template you have Movable Type create a sub-directory with the same name and name the file created with 2011-09.html. That is nice since it will identify the "issue" of the home page the file represents. So you use the template output option in the cloned template to make a new folder as it makes the file, such as 2011-09/2011-09.html.

Into that same sub-directory, lets say Archives/2011-09/ you put a individual style sheet. You can create this with a clone of the Movable Type installed stylesheet styles.css, which you should have renamed Archive.css when you created this Archive blog sub-directory. You can link to itself in the template file, so that way when you use Stylizer to change the look of the page, you can save the css to the website and the next time you publish the 2011-09css it will see that the time stamp of the existent file is newer and suck that into the Movable Type database and make it the new 2011-09.css template.

Linking a template to itself can work for the monthly issue html, but here is the rub, you can't have any Movable Type XML tags in the file. Those tags are meaningless to a browser trying to display the page. So the page you author in an outside program like Komposer has to be straight html with no tags. That means if your homepage needs tags to build itself, you are best just trying to write the page in Movable Type's template editor.

This is so miserable I have pretty much given it up. I just have a homepage that has real links to content as opposed to those auto-generated links that programmers love to use Movable Type tags to generate. The home page is way too important to design in some command line linear geek method. You have to be like an art director when you design a home page. I like to author it in Komposer 7, using Stylizer to play with formatting in the stylesheet. I want each new issue to have a big bold background picture, so that also lives in the 2011-09 sub-directory or whatever issue date I am working on.

Now that is not to say you don't use the tags and database of Movable Type to help organize all these home pages. The reason you give the month issue template pages a uniform name like 2010-11 is so Movable type can sift through them and present them in a nice order in the Archive directory index.html file. I also rig it so that there is a thumbnail of the home page displayed in the Archive/index.html file. I do this by taking a screen grab of the finished page in my browser, then paste it into IrfanView, where I can trim and re-size the home page to 300px wide. I save that with the same naming convention, 2011-09.jpg, in the monthly issue sub-directory, 2011-09/ in this case.

Although this was intended to allow me to publish monthly issues, it was also a great way to show old versions of the website before I starting doing this "issue" paradigm. Once you build up one or more sub-directories with the homepages, css, local images in them, as well as that screen-shot thumbnail, you can play with Movable Type tags in the Archive/index.html template file.

To start out, I used the following tags inside the Archive/index.html template:

```
<h1>Home pages from the 1990s</h1><br>
<mt:IndexList>
  <mt:If tag="IndexName" like="^1">
     <a href="<$mt:IndexLink$>">
<$mt:IndexName$></a> back issue
<br>
  </mt:If>
</mt:IndexList>
<br>
  <h1>Home pages from the
2000s</h1>
<br>
<mt:IndexList>
  <mt:If tag="IndexName" like="^2">
    <img src=
"<$mt:IndexName$>/<$mt:IndexName$>.jpg"
alt="Cover" /> <a href="<$mt:IndexLink$>">
<$mt:IndexName$></a> back issue
<br>
  <br>
  </mt:If>
</mt:IndexList>
<br>
```

I had to mess with the
 tags in order to the the spaces I wanted, it seemed a carriage return to isolate it on a line finally got it to render a space in the page. The Movable Type tags combined with plain characters:

```
<$mt:IndexName$>/<$mt:IndexName$>.jpg
makes the string
2011-09/2011-09.jpg
```

to call out the thumbnail form its sub-directory as the script loops through all the index template names. The goofy like="2" was the only way I could get the entries listed that started with "2" like 2011-09.

The template loops through creating html until it runs out of index names starting with 2. There are a million way to do this and to order the tags. I chose by decade since it was easy to parse against a 2 or a 1. I will have to think what I will do not that it is 2011. I guess the "like" would be 200 and 201 to get those in separate decades.