How to build a blog with Adobe Muse and Business Catalyst Workbook.

V1.0

Authored by Chris Kellett museGrid.com/Simple Flame

Site uses the Mavens Coffee Shop Template from museGrid.com
Contents

Introduction 3

About Business Catalyst. 3

About the Author of this workbook, Chris Kellett. 3

What we will learn. 4

Prerequisites and conventions. 5

What you will need... 5

Text conventions in this document. 6

The Project Template 7

What will the finished blog look like? 7

Understanding the Business Catalyst Blog Module. 8

The Build 9

PART 1: The Overall Blog Layout 9

PART 2: The Blog Post List Layout 15

PART 3: The Blog Post Details Layout 20

Publishing the site to (BC) and making edits. 24

Creating a blog in (BC) 25

Modifying your Layouts in (BC) 27

The Blog Post Layout (BC) 27

The Blog Post List Layout (BC) 28

The Overall Blog Layout (BC) 29

Modifying the CSS (to make sure it doesn’t all break) 30

Baking the code to avoid edit overwrites. 32

What we have learned. 34

Taking it further - adding a latest articles section to the home page. 35

Building the Latest Blog Posts Layout. 35

Adding the (BC) code. 39
Introduction

Adobe Muse is an amazing tool for quickly building stunning websites with no code. These sites are feature rich, standards compliant and reactive to tablet and mobile devices.

Muse however is not a visual Dreamweaver but rather a layout package more akin to inDesign which means its not as straight forward to build dynamic sites with data driven content as it would be in Dreamweaver. That being said with a little planning and understanding you can achieve amazing dynamic results with Muse on the Adobe Business Catalyst platform.

About Business Catalyst.

Adobe Business Catalyst is an all-in-one hosting platform that includes powerful modules for news items, custom built web apps (think a purpose built directory or real estate listings) and of course a blogging engine.

BC (Business Catalyst) uses a unique system of tags to reference data from the database and modules a collection of tags within an HTML layout. This allows you to easily integrate BC content with an HTML page without the need for .php or .aspx extensions to pages.

The great news for you and I is that Muse creates .html pages so we can easily embed BC tags and modules into our sites which opens an entirely new way of using Adobe Muse to create highly attractive and functional websites.

About the Author of this workbook, Chris Kellett.

I am designer/developer and a very active Muse and BC user. Until the beginning of 2013 I worked as a freelancer for 14 years working on print, web and film projects throughout the UK and parts of the US.

In early 2013 I started a new career with the amazing guys at Simple Flame where we setup museGrid.com the number one location for business quality Muse Templates and Widgets (We also offer FREE video based training for Muse).

I have built over 40 sites and templates between Q4 2012 and Q1 2013. I am a regular on Muse Jam sessions with Dani Beaumont. I spend most of my working life nowadays within Muse.

I am also the guy that figured out how to properly integrate BC with Muse so you are in safe hands.
What we will learn.

In this session we will learn how to build a blog layout in Adobe Muse, upload it to Business Catalyst and install the tags and modules into our HTML to make the blog work and look great.

We will see how we can “Bake” the code into BC so that future publishing actions from Muse to your website will not overwrite the modifications we make.

Hopefully, you will learn the thinking behind the process to enable you to use the techniques learned to work with your own layouts and more of the BC modules and functionality.

Most importantly have fun.

**NOTE:** Adobe Muse is aimed at building websites without the need for code which it does admirably. However, when you start looking at integrating code elements into the HTML there is a need to deal with the code itself in small doses. The process is not difficult but could be daunting if you have not done this before. We aim to give you as much direction as possible.
Prerequisites and conventions.

What you will need...

To get started with this project you will need the following.

1. **The Mavans Coffee Shop Files.** In the files package you will find all the assets for the project along with multiple .muse versions of the Mavens Coffee Shop Website. Each version is a different stage of the build process which allows you to go from the initial starting point to the completed site ready for publishing to Business Catalyst.

   We have also included the complete files that will reside on business catalyst that can either be used as a reference to check your code or simply uploaded to test the process.

2. **Business Catalyst Trial or Full Account** - As we are using the blogging tools built into BC you will need access to a BC trial account (If you are a partner you will know how to do this) if not just go to Business Catalyst ([http://www.businesscatalyst.com/](http://www.businesscatalyst.com/)) and click on the “Get started for free” button.

   Select the blank template and follow the onscreen instructions. Once you have received your trial account setup email you will be able to publish the Muse files to this trial account for testing and previewing.

3. **The latest version of Adobe Muse installed on your computer.** Build 4.1 build 8 or above.

4. A coffee, comfy seat and a desire to explore new ideas.
Text conventions in this document.

We will use the following text convention in this document to help you identify various types of code and content.

When discussing HTML or CSS code such as `<div class="style">` it will be displayed as `<div class="style">` or this `.class { margin-left: 10px}`

Business Catalysy specific tags and modules will be shown as `{tag_thistag}` or `{module_thismodule,a}`

**NOTE:** Tips and notes will be shown like this

**WARNINGS:** Will look like this.
The Project Template

For this project we will be using the Mavens Coffee Shop template from museGrid.com. This is a single page layout design which we have added a link to the blog in the main navigation.

The site has a dedicated mobile version but we will focus on the desktop/tablet version.

The principals learned in this exercise are intended to equip you to build a blog in your own projects so try and grasp the concepts and workflows rather then focusing on the actual layout and features of this particular template.

What will the finished blog look like?
Understanding the Business Catalyst Blog Module.

First lets take a look at the structure of the blog. The blog structure is as follows, the page template is the starting point inside of which is the overall blog layout which may contain the sidebar with blog tags, categories and so on.

The blog post list also sits inside the overall blog layout. Clicking an individual blog post links you to the blog post details page. So in effect we have four separate pieces of content to think about.

1. The page template.
2. The **Overall Blog Layout** which contains the side bar for categories, archive etc.
3. The **Blog Post List Layout** which contains the data for each individual blog post in a list format to allow the page to display multiple posts.
4. The **Blog Post Details Layout** which is the full detail of an individual blog post.

Below is a visual representation of the blogs structure.

NOTE: We are using the Mavens Coffee Shop template from museGrid.com so we already have “Page Master/Template” so we only need to make the other three parts.
The Build

Now we have all the principles in mind, we know what the final build will look like lets build this step by step.

PART 1: The Overall Blog Layout

**NOTE:** If you would like to skip the process of setting up the layouts and jump straight to the publishing process open the Muse file “mavens-with-blog.muse”

The first item to build is the overall blog layout. This will sit inside of the page template.

1. First create a new page in the Muse Plan view and call it “Blog Overall Layout” make sure you apply the “Page Master” to this page.

2. The blog heading - Use paragraph style **H1 Section Headings** and **align the text right.** Use the Transform settings in the image.

3. The RSS feed icon - Create a rectangle with the transform settings below.
   In the states panel select “Normal” and “Active” to background fill to an image ([images > rssA.png](images > rssA.png))

   For the “Rollover” and “Mouse Down” states use the background fill and choose an image ([images > rssB.png](images > rssB.png))

   In the link section just type the word “link” and hit return this will ensure that once the file is uploaded it already has the appropriate HTML to allow us to easily add the correct RSS feed link to the icon.
4. The blog post area - This is where we will place the “Module” code in Business Catalyst to render the individual blog posts in list format onto the page.

**NOTE:** Change the text color to a lighter color so you can visually see where it is.

5. Recent Posts - This has two parts to it the title and the tag. We will replace the tag in Business Catalyst with the correct (BC) tag which will render content when the page loads in the browser.

   Title: Use the paragraph style **Blog Sidebar Heading** and apply the graphic style **Blog Sidebar Title** to the same unit. This will add the Font style and line as shown here.
Tag: Use the paragraph style **Blog Sidebar Link Text** whilst the item is still selected apply text link style **Blog Sidebar Link** as well (from the “Hyperlinks” menu) this will ensure that the correct link and rollover styles are used when the (BC) tag is added.

---

6. Blog Archive - This will show the blog posts archived into months. This has two parts much like the “Recent Posts” seen previously.

Title: Use the paragraph style **Blog Sidebar Heading** and apply the graphic style **Blog Sidebar Title** to the same unit. This will add the Font style and line as shown here.

---

Tag: Use the paragraph style **Blog Sidebar Link Text** whilst the item is still selected apply text link style **Blog Sidebar Link** as well (from the “Hyperlinks” menu) this will ensure that the correct link and rollover styles are used when the (BC) tag is added.
7. Tag - This has two parts much like the “Recent Posts” and “Blog Archive” seen previously.

Title: Use the paragraph style **Blog Sidebar Heading** and apply the graphic style **Blog Sidebar Title** to the same unit. This will add the Font style and line as shown here.

Tag: Use the paragraph style **Blog Sidebar Link Text** whilst the item is still selected apply text link style **Blog Sidebar Link** as well (from the “Hyperlinks” menu) this will ensure that the correct link and rollover styles are used when the (BC) tag is added.

8. Pagination - This allows you to navigate through pages of posts and will show “Next” and “Previous” links as well as clickable page numbers.
NOTE: We need to place a rectangle behind the pagination to ensure correct positioning and wrapping of the content once the page has been uploaded and modified in (BC).

Previous Page Link: Use paragraph style **Pagination** whilst the item is still selected apply text link style **Pagination** as well (from the “Hyperlinks” menu) this will ensure that the correct link and rollover styles are used when the (BC) tag is added.

![Previous Page Link](image)

Page Numbers Link: Use paragraph style **Pagination** whilst the item is still selected apply text link style **Pagination** as well (from the “Hyperlinks” menu) this will ensure that the correct link and rollover styles are used when the (BC) tag is added.

![Page Numbers Link](image)

Next Page Link: Use paragraph style **Pagination** whilst the item is still selected apply text link style **Pagination** as well (from the “Hyperlinks” menu) this will ensure that the correct link and rollover styles are used when the (BC) tag is added.
Background Rectangle: Set its fill color to transparent and its border size to zero [0].

That's the Overall Blog layout now complete and ready to publish.

Next we will create the Blog Post List Layout.
PART 2: The Blog Post List Layout

For this part of the project we need to get our heads around how this is going to work. We know that the Overall Blog Layout sits within the page itself. Well the Blog Post List sits inside the Overall Blog Layout so we don’t need the page template or Overall Blog Layout on this page.

When the site is published and we have incorporated all of the correct (BC) tags and modules in place the page will render with the Page Layout on the outside, the Overall Blog Layout within the Page Layout and the Blog Post List inside of the Overall Blog Layout a bit like a Russian Doll.

So the Blog Post List page needs to be a little different to all the rest. For example we do not need the page to centre as it is inside the page which itself is centered. So we need it to be set top/left alignment.

We don’t want and margins or padding either as we need it to sit flush in its content holder.

So the workflow for setting up a “Module Layout” is to first create the “Module Master” in Muse and then add the content.

1. The “Module Master” - Create a new “Master Page” in the Muse Plan view with the following page property settings (Page > Page Properties)

   Next open the master and set the “Browser Fill” to a dark color from the color picker. This will allow us to see the content on the page easier. Set the “Fill” colour to transparent and the border to zero[0].

   Make sure all of the page head and footer guides are fully to the top and bottom of the
2. Next in Muse Plan view under the “Blog Overall Layout” page add two new pages called “Blog Post List” and “Blog Post Details” apply the “Module Master” to both pages.

We are now ready to add the layout for the Blog Post List and the Blog Post Detail after which we will be ready to publish.
3. Blog Post List : Title and Date - Use the paragraph style **Blog Post Title** making sure the text is aligned left whilst the item is still selected apply text link style **Blog Post Title Link** as well (from the “Hyperlinks” menu).

4. Blog Post List : Post Body Content - Use the paragraph style **Blog Post Text**.

5. Blog Post List : Read Blog Post Button - Create a rectangle with the transform settings below.

In the states panel select “Normal” and “Active” to background fill to an image (images > read-blog-post.png) and set the position to top/left as below.
For the “Rollover” and “Mouse Down” states use the background fill and choose an image (images > read-blog-post.png) and set the position to bottom/left as below.

In the link section just type the word “link” and hit return this will ensure that once the file is uploaded it already has the appropriate HTML to allow us to easily add the correct link to the icon.

6. Blog Post List : Permalink - Use the paragraph style Permalink Text whilst the item is still selected apply text link style Permalinks as well (from the “Hyperlinks” menu).

7. Blog Post List : Blog Post Separator Line - Create a rectangle as below and apply the graphic style Blog Post End Line.

8. Blog Post List : Background Rectangle - Create a rectangle and arrange to “Send To Back” so that it sits behind the content. This will ensure that each post will sit inside a rectangle and will be aligned correctly. Set its fill color to transparent and its border size to zero [0].
Blog Post List: Background Rectangle (Continued)

{Blog Post Title} I {Date here}

(This is where the body text for the blog post.)

read blog post

Get the {Permalink}
PART 3: The Blog Post Details Layout

1. Blog Post Details: Title, Date and Post Body Content - Copy and paste those items from the Blog Post List page to speed up the process. Use the (Edit > Paste In Place) option to get the items to sit in exactly the same place as you copied them from.

2. Blog Post Details: Separator Line One - Create a rectangle as below and apply the graphic style *Blog Post End Line*.

3. Blog Post Details: Separator Line Two - Create a rectangle as below and apply the graphic style *Blog Post End Line*.

4. Blog Post Details: Previous Page Button - Create a rectangle with the transform settings below.
In the states panel select “Normal” and “Active” to background fill to an image (images > previous-page.png) and set the position to top/left as below.

For the “Rollover” and “Mouse Down” states use the background fill and choose an image (images > previous-page.png) and set the position to bottom/left as below.

In the link section add this javascript code `javascript:history.back()` and hit return this will give the button the functionality to go back one page when the website is published.

5. Blog Post Details : Trackbacks & Permalinks - Use the paragraph style Permalink Text whilst the item is still selected apply text link style Permalinks as well (from the “Hyperlinks” menu). Make sure the text is aligned left.

6. Blog Post Details : Track Back Link - This is made up of two parts the title and the tag.

   The Title - Use the paragraph style Permalinks Text and make sure it is aligned left.
The Tag - Use the paragraph style **Track Back Text** whilst the item is still selected apply text link style **Blog Post Title Link** as well (from the “Hyperlinks” menu). Make sure the text is **aligned left**.

7. Blog Post Details : Track Backs - We are basically replicating what we have just done with the “Track Back Link” with the following transform settings:

The Title - Use the paragraph style **Permalinks Text** and make sure it is **aligned left**.

The Title - Use the paragraph style **Track Back Text** whilst the item is still selected apply text link style **Blog Post Title Link** as well (from the “Hyperlinks” menu). Make sure the text is **aligned left**.
8. Blog Post Details: Background Rectangle - Create a rectangle and arrange to “Send To Back” so that it sits behind the content. Set its fill color to transparent and its border size to zero [0].

We now have all the build elements in place to be able to publish the site and start editing in Business Catalyst.

**WARNING:** Ensure that you have everything setup correctly before publishing. Any changes you make to the code on the Business Catalyst server will be overwritten each time you publish the site from Muse.

Once the Business Catalyst work has been completed we will take precautions to ensure that overwriting can't take place but just bare that thought in mind.
Publishing the site to (BC) and making edits.

So we have all of the layouts ready now we are ready to publish the site. We will then login to our Business Catalyst control Panel and add the modules and tags to connect the layouts to the data.

As you know and we have mentioned before Muse is web design without code, however when you add data into the mix you have to at least be prepared to get your hands a little bit code dirty. We will only be modifying, copy and pasting with a little bit of code writing but don't worry its super easy. Lets get started.

Assuming you have done any of the following:

1. You are already a partner and have set up a trial site for this project.

2. Have gone to www.businesscatalyst.com and started a free trial site using the “Blank Template” option.

Go ahead and hit the “Publish” button and let the site upload.

3. When the site has fully uploaded click the “Manage” button in Muse this will log you into your Business Catalyst site.

NOTE: Use the “Manage” button rather then navigating to the sites admin area in your browser to ensure any further publishing of the site does not log you out of the admin area.
Creating a blog in (BC)

We need to setup a blog and import or add some content to that blog to ensure we can test and preview the site correctly.

1. Click on the Modules > Blogs in the left hand dash area. Click on the “Blogs” option then on the “AddBlog” button.

2. Use these settings.

NOTE: You can use whatever details you like when you build your own site for the purpose of this exercise please use these exact settings. Remember the URL will be different to this image.
3. Next we are going to import some blog posts so that we have some blog data to preview. (BC) has a great tool for importing blog posts from other services so we are going to select the Wordpress option.

Now we have our blog setup, we have blog posts we can start adding the data tags and modules to the layouts we generated and published in Muse.

4. If you wish to preview what the blog currently looks click the preview button.
Modifying your Layouts in (BC)

Now comes the part you have been waiting for editing the files and installing the Business Catalyst Modules and Tags.

We will be using the (BC) “Develop” panel extensively for doing our editing work. The develop tool built into (BC) is based on the Edge Code/Brackets tools and is an exceptional code editing tool (especially one that works in the browser).

We are going to work backwards with our code editing we will start with the Blog Details Layout then the Blog Post List and finally the Overall Blog Layout. We are doing this to ensure that everything is complete before seeing the preview.

The Blog Post Layout (BC)

1. Open the page for editing in “Develop” choose the “blog-post-details.html” page. Copy all of its contents to your clipboard and close the file.

2. Next open the file Layouts > Blog > post.html select all the code and delete it. Then paste the code you just copied into it.

3. Now we are going to edit the code find the line that reads <div class="clearfix" id="page"> and change it to <div class="clearfix blogBox">

Next we are going to start adding the (BC) data tags. In the “Toolbox” on the right hand side of the page click on the “Data” button. You will then see a list of all the data tags available.

In the code find the text that says {Blog Post Title} highlight it and then click on the {tag_blogposttitle}
4. Go through the same process for the following:

<table>
<thead>
<tr>
<th>{Date here}</th>
<th>Becomes</th>
<th>{Date here}</th>
</tr>
</thead>
<tbody>
<tr>
<td>{This is where the body text for the blog post.}</td>
<td>Becomes</td>
<td>{tag_blogpostbody}</td>
</tr>
<tr>
<td>{0}</td>
<td>Becomes</td>
<td>{tag_trackbackcount}</td>
</tr>
<tr>
<td>{Permalink}</td>
<td>Becomes</td>
<td>{tag_permalink}</td>
</tr>
<tr>
<td>{Track back link}</td>
<td>Becomes</td>
<td>{tag_trackbacklink}</td>
</tr>
<tr>
<td>{Track back link}</td>
<td>Becomes</td>
<td>{tag_trackbacklist}</td>
</tr>
</tbody>
</table>

Save the document and then close it.

**The Blog Post List Layout (BC)**

1. Open the page for editing in “Develop” choose the “blog-post-list.html” page. Copy all of its contents to your clipboard.

2. Next open the file Layouts > Blog > postlist.html select all the code and delete it. Then paste the code you just copied into it.

3. Now we are going to edit the code find the line that reads

```html
<div class="clearfix" id="page"> and change it to <div class="clearfix blogBox">```

Next we are going to start adding the (BC) data tags. In the “Toolbox” on the right hand side of the page click on the “Data” button. You will then see a list of all the data tags available.

In the code find the text that says `{Blog Post Title}` highlight it and then click on the `{tag_blogposttitle}`
4. Go through the same process for the following:

<table>
<thead>
<tr>
<th></th>
<th>Becomes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>{Date here}</td>
<td>{tag_blogpostbody}</td>
<td>{Date here}</td>
</tr>
<tr>
<td>{This is where the body text for the blog post.}</td>
<td>Becomes</td>
<td>{tag_itemurl_nolink}</td>
</tr>
<tr>
<td><a href="http://link">http://link</a></td>
<td>{tag_itemurl_nolink}</td>
<td></td>
</tr>
<tr>
<td>{Permalink}</td>
<td>Becomes</td>
<td>{tag_permalink}</td>
</tr>
</tbody>
</table>

Save the document and then close it.

**The Overall Blog Layout (BC)**

1. Open the page for editing in “Develop” choose the “blog-overall-layout.html” page. Copy all of its contents to your clipboard.

2. Next open the file Layouts > Blog > overall.html select all the code and delete it. Then paste the code you just copied into it.

3. In the code find the text that says href="index.html#home" change it to href="/index.html#home" do the same with any similar links you find on the page putting a forward slash at the beginning of the link reference as in the list below:

<table>
<thead>
<tr>
<th>index.html#anchor-u414</th>
<th>Becomes</th>
<th>/index.html#anchor-u414</th>
</tr>
</thead>
<tbody>
<tr>
<td>index.html#about</td>
<td>Becomes</td>
<td>/index.html#about</td>
</tr>
<tr>
<td>index.html#rewards</td>
<td>Becomes</td>
<td>/index.html#rewards</td>
</tr>
<tr>
<td>_blog/BLOG</td>
<td>Becomes</td>
<td>_blog/BLOG</td>
</tr>
<tr>
<td>index.html#contact</td>
<td>Becomes</td>
<td>/index.html#contact</td>
</tr>
<tr>
<td>privacy.html</td>
<td>Becomes</td>
<td>/privacy.html</td>
</tr>
</tbody>
</table>

Why are we doing this? Well now that we have posted the code into a file that resides deeper within the site file structure the links on the page would break. By putting a forward slash in front of the links it will link to the root files.

**NOTE:** You can search in the code panel click your cursor into the code anywhere and press Command > F to open the search dialog.
4. In the code find the text that says {Replace this with the blog module} highlight it and then click on the {tag_postlist,3} (The number denotes how many posts will be shown per page.

<table>
<thead>
<tr>
<th>Recent post tag</th>
<th>Becomes</th>
<th>tag_blogrecentpost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blog Archive tag</td>
<td>Becomes</td>
<td>tag_blogpostarchive</td>
</tr>
<tr>
<td>Tag List tag</td>
<td>Becomes</td>
<td>tag_blogtaglist</td>
</tr>
<tr>
<td>Previous</td>
<td>Becomes</td>
<td>tag_previouspage</td>
</tr>
<tr>
<td>Page Numbers</td>
<td>Becomes</td>
<td>tag_pagination</td>
</tr>
<tr>
<td>Next</td>
<td>Becomes</td>
<td>tag_nextpage</td>
</tr>
</tbody>
</table>

Line 28 - class="clearfix colelem" Becomes class="clearfix colelem Blog-Post-Text"

Save and close the file.

Modifying the CSS (to make sure it doesn’t all break)

If you previewed the site right now it would appear a little broken. This is because we need to modify some of the CSS associated with the files. At this point you need to get your head around what we are doing here each page uploaded from Muse is connected to a set of style sheets the global stylesheet that contains all of the paragraph, character and graphic styles along with styles that are shared across the entire website.

Each page also has its own unique stylesheet that contains the styles control the individual page layout information. As it is machine generated code some of the style references are the same for each page so if you are including one page within another which is what we will be doing in rendering the page with the blog post list in it there will be conflicting styles which break the page.

Also in Muse we were not able to add so of the target HTML tags to a particular style so we need to do that manually in the code which is what we will do now.

1. Open the file CSS > site_global.css find the line 404 .Bodytext--Verdana,.Blog-Post-Text and add a space and the letter “p” to it so it read .Bodytext--Verdana,.Blog-Post-Text p

   This ensures that the style now targets the paragraphs or “<p>” tags within the blog post to correctly format.

2. Next find the line 413 .Blog-Sidebar-Link-Text and add a space and the letters “li” to it so it read .Blog-Sidebar-Link-Text li

   This ensures that the style now targets the paragraphs or “<li>” tags within the blog post to correctly format. Save and close the file.
3. Next open the file CSS > site_global.css remove all of the following code:

```
html {
  background-color: #2C0A08;
}

#page {
  z-index: 1;
  width: 648px;
  min-height: 154.999999999999px;
  padding: 1px 311px 44.00000000000114px 1px;
}
```

Replace with:

```
/* This has been changed to allow stacking of posts */

.blogBox {
  z-index: 1;
  width: 700px;
  min-height: 200px;
  /* this has been modified to 0 */
  padding-right: 0px;
  margin-bottom: 60px;
}
```

Save and close.

4. Next open the file CSS > site_global.css remove all of the following code:

```
html {
  background-color: #2C0A08;
}

#page {
  z-index: 1;
  width: 649px;
  min-height: 114px;
  padding: 1px 310px 85px 1px;
}
```

Replace with:

```
/* This has been changed to allow stacking of posts */

.blogBox {
  z-index: 1;
  width: 700px;
  min-height: 200px;
  /* this has been modified to 0 */
  padding-right: 0px;
  margin-bottom: 60px;
}
```

Save and close.

Preview the site and see your lovely new blog designed in Muse, modified in (BC).

How awesome is that? Now lets bake the code to stop future publishes from Muse overwriting the site.
Baking the code to avoid edit overwrites.

1. In the “Develop” console click on the stylesheets folder

Whilst selected click on the add new page icon,

Call the new page “blog-global.css” hit the return key to save the page. Open the file
CSS > site_global.css copy its contents and paste it into the new “blog-global.css”. Save and close both follow the same process for the following pages.

Create a new page in the “stylesheets” folder call it “blog-overall.css” copy the contents of CSS > blog-overall-layout.css into it. Save the file and close them both.

Create a new page in the “stylesheets” folder call it “blog-postlist.css” copy the contents of CSS > blog-post-list.css into it. Save the file and close them both.

Create a new page in the “stylesheets” folder call it “blog-post-details.css” copy the contents of CSS > blog-post-details.css into it. Save the file and close them both.

This means we now have the code we modified in (BC) that controls the the blog page layouts in a place where Muse publishing will not touch it. Next we need to change the references to the CSS files in the layouts to point to these new files.

2. Open the file Layouts > Blog > overall.html on the line 11 change the stylesheet files references from “/stylesheets/blog-global.css?368364125” to “/stylesheets/blog-global.css”

On line 14 change the reference from “/css/blog-overall-layout.css?4004737374” to “/stylesheets/blog-overall.css”

Save and close.

3. Open the file Layouts > Blog > postlist.html on the line 9 change the stylesheet files references from “/css/site_global.css?368364125” to “/stylesheets/blog-global.css”

On line 14 change the reference from “/css/blog-overall-layout.css?4004737374” to “/stylesheets/blog-postlist.css” also remove the code id="pagesheet" from this line.

Save and close.
4. Open the file **Layouts > Blog > post.html** on the line 9 change the stylesheet files references from “/css/site_global.css?368364125” to “/stylesheets/blog-global.css”

On line 14 change the reference from “/css/blog-post-details.css?306356067” to “/stylesheets/blog-post-details.css” also remove the code id="pagesheet" from this line.

Save and close.

Now if you ever republish the site from Muse your blog code changes will not be affected.
**What we have learned.**

We have learned to build layouts in Muse using our specially setup “Modules Master” to create our various module layouts.

We have produced our layouts in such a way that they are easy to add tags and modules to within the (BC) “Develop” panel.

We have learned that we need to change certain CSS properties to ensure that nested pages work properly.

We have learned to bake our code out so that when we publish the site in future, Muse will not overwrite the files we have modified.

We have learned that now you can build anything with Muse and Business Catalyst in a fraction of the time it would take to hand code the site.
Taking it further - adding a latest articles section to the home page.

To further expand on this project we will be adding a custom layout to the front page of the website. This will allow us to show the most recent article from the blog on the main landing page of the site.

Building the Latest Blog Posts Layout.

1. Create a new page in the Plan view of Muse under the Blog Post List Layout page. We are adding it underneath for more of a visual representation of the site structure then anything else as it allows us to see how the site and layouts are connected.
2. Create a rectangle with the fill color of \#D62545 set the border width to [0] switch on all four radius buttons and set the radius size to 100.

Go to the states panel and set the “Normal” and “Active” states to a color fill of \#D62545

Set the “Rollover” and “Mouse Down” states to a color fill of \#28BFAD

3. Next add the text as bellow and use the paragraph style Blog Latest Date (Day) make the text alignment centered.
4. Next add the text as below and use the paragraph style Blog Latest Date (Month) make the text **alignment centered**.

5. Next add the text as below and use the paragraph style Blog Latest Date (Month) make the text **alignment centered**.

6. Next add the text as below and use the paragraph style Blog Latest Updates Title whilst the item is still selected apply text link style Blog Sidebar Link as well (from the “Hyperlinks” menu). Make sure the text is **aligned centered**.
7. Finally add a rectangle send it to the back of the page (Object > Send To Back) and add the word “link” to the rectangle in the link box ensure you hit return once added to ensure the link is applied to the rectangle. This will make sure that the layout is ready to have the link applied to it in (BC)

8. Now we want to add the module code to the front page of the website.

Go to the “Home” page of the site and find the “Latest Updates” area of the page adding the text box as below.

This page is now ready to upload using the publish button.

**WARNING:** Remember that Muse will overwrite any changes you have made on the website to other pages unless you have taken the precautions mentioned in the “Baking the code” section of this document.
Adding the (BC) code.

We are going to be doing the same processes that we have already learned to finish the project of and add the latest blog posts to the home page of the site.

1. Login into the (BC) admin area of the site and click on the “Develop” button.

• Open the page latest-blog-post.html swap out the following code with tags:

   <div class="clearfix id="page"> becomes <div class="clearfix LatestItem">
   {Blog Post Title Here} becomes {tag_blogposttitle}
   DAY becomes {tag_blogpostday}
   MONTH becomes {tag_blogpostmonth}
   YEAR becomes {tag_blogpostyear}
   http://link becomes {tag_itemurl_nolink}

   Save and leave open. Copy the contents of this file.

   Click on the “ModuleTemplates” folder and create a new folder called “Blog”.

   Click on the new “Blog” folder you just created and create a new file called “latestBlog.tpl” and paste all the contents from the “latest-blog-posts.html” file into it.

   Save the file and close all open files.

   Open the file CSS > latest-blog-posts.css and do the following:

   Remove...

   .latest-item
   {
     z-index: 1;
     width: 200px;
     min-height: 200px;
     float: left;
     margin-right: 10px;
   }

   Replace with

   .LatestItem
   {
     z-index: 1;
     width: 960px;
     min-height: 200px;
   }

   Save leave open and copy the code.

   Now let’s bake that code. Create a new file in the stylesheets folder called “blog-latest.css” and add the code from CSS > latest-blog-posts.css to it hit return and save.

   Open the “latestBlog.tpl” file and re-reference the css files to those in the stylesheets
folder.

/css/site_global.css?368364125 becomes /stylesheets/blog-global.css

/css/latest-blog-posts.css?3976389184 becomes

Remove the code id="pagesheet".

Save and close.

2. Next open the page “index.html” swap out the following code:

{Latest blog items module goes here} and replace it with

{module_blogsitepost,4 template="/ModuleTemplates/blog/latestBlog.tpl"}

Save and close.

NOTE: You can search in the code panel click your cursor into the code anywhere and press Command > F to open the search dialog.

Preview the site and hey presto you have a cool latest updates panel with a unique style to match the site. Designed in Muse connected in (BC).
Thanks for joining in on this session we hope it has inspired you to take a further look at using Muse to build sites for use on Business Catalyst.

You can contact me at Simple Flame/museGrid.com

chris@simpleflame.com

Feel free to drop me a line or send me a link to your new Muse projects and see you at the next Adobe Muse Jam.