Joomla 3.0 And Beyond

With the release of Joomla 3.0 we wanted to update everyone on the changes within Joomla, modifications to Shape5 products, and our recommendations for moving forward.

Should I Upgrade to Joomla 3.0?

This is the question on everyone's mind right now. The simply answer is, no! Joomla 3.0 is strictly for developers and advanced users. If you are currently using 2.5, stay right where you are and do not upgrade until at least 3.5! Unless you are a very experienced user or developer we also recommend still creating even new websites with 2.5 since it is a more stable platform with many more extensions available for it. These are recommendations of Joomla themselves and we agree with them. You can see Joomla's recommendations and timeline of future releases at the following url:

http://community.joomla.org/blogs/community/1682-joomla-25.html

http://www.youtube.com/watch?v=LPcp7rUeidM

Will My 2.5 Product Work With 3.0?

It is true that Joomla itself will upgrade with a simply click of a button through Joomla's internal upgrade component, but that does not mean that templates and extensions will work with the new version. We cannot stress enough just how different the core functions of Joomla 2.5 and 3.0 are. Many core functions have been renamed or removed, as well as class names and id's used for css styling. The following url will give you a relatively complete list of all the changes made in 3.0, however there are more class names than what are listed here:

http://docs.joomla.org/Potential_backward_compatibility_issues_in_Joomla_3.0_and_Joomla_Platform_12.1

Vertex 3 and Jquery

One of the biggest changes in Joomla 3.0 is the introduction of jquery. Mootools is still included with 3.0, but without mootools 1.2 support anymore. It is our opinion that running two javascript libraries on the same site is not a good idea and will effect the performance of the site. Since joomla has begun the transition over to jquery so will Shape5 products. Even though mootools is still included in 3.0, it is our general assumption that it will soon be removed, maybe even as soon as 3.5. Rather than waiting until mootools is removed altogether we have decided to make the switch now. This means that scripts like S5 Flex Menu, Multibox, Lazy Load, etc. are all currently being re-written from mootools into jquery. In addition to the switch to jquery we have had to make several other updates to Vertex so that everything integrates correctly with the new function names of 3.0. This is a very in depth project and will take several weeks to complete. This update will apply to both 2.5 and 3.0 Vertex templates, which means jquery will also be used on 2.5 templates as well once our new Vertex patch is released.

Will There Be Multiple Versions of Products Again?

Shape5 has been around since Joomla 1.0 and we know the confusion and frustration that multiple versions of products often cause. Unfortunately, due to the many changes in the core Joomla functions many of the products will need to be updated again to work with 3.0, just as they were from 1.0 to 1.5 and 1.5 to 1.6. Vertex templates will simply be updated

with the Vertex patch mentioned above, and possibly some css additions. Our goal as we update each product is to create a single file that works with both 2.5 and 3.0+ using JVersion. This will ensure that multiple versions are not needed, this will definitely be true of Vertex templates. Creating a Vertex patch is our first priority, and we will then beging converting older templates and extensions.

Because our goal is to release only one version of each template that means we will also be releasing only one site shaper. We will package our shapers with Joomla 2.5 for quite some time. Those who are interested in upgrading to 3.0 can do so in the backend of Joomla after installation.

How Will Shape5 Incorporate Bootstrap?

This is really a two part answer. The first part of Bootstrap is a vast array of typography items, buttons, tabs, menus, etc. The other part of Bootstrap is the layout itself and it's responsive media queries. We will address each of these below individually.

Typography - The goal behind adding Bootstrap to Joomla 3.0 is to allow extension developers to all have a common framework to build off of in regards to buttons, menus, typography, etc. This allows template developers to simply stylize certain class names rather than supply specific css for every component and extension individually. It also helps extension developers focus more on coding rather than stylizing. It is a much more centralized and organized way of handling the appearance of each extension. This approach also makes it much easier for the end user since all extensions will have a common look and feel to them. Some of these Bootstrap elements can be found here:

http://twitter.github.com/bootstrap/base-css.html

http://twitter.github.com/bootstrap/components.html

These items will be included in our templates simply by calling the Bootstrap js and css files that will now come with the Joomla core. This procedure is described here:

http://docs.joomla.org/Converting_a_template_for_Joomla_3

Layout - One of the biggest buzz words about Joomla 3.0 is "Bootstrap", when in all reality the buzz word should be "responsive". There is nothing in Joomla 3.0 that requires the Bootstrap layout to be used in a template, it's the css and js portion mentioned above that should be called by templates to allow extensions to perform properly and be stylized correctly. The layout used in Bootstrap is not a requirement. The boostrap layout is a row and column grid system based on percentage widths. Each row consists of 12 sections, and each div used in the row contains a class name that specifies how many sections that particular div will take up in width. The total number of sections used in every row, through class names, must total to 12. This does not mean that there are 12 divs on every row, it's simply how the layout determines what percentage to apply to each div in that particular row, and that is done through class names with pre-determined widths applied to them via css. For example, if I told a div to stretch 2 sections, then it's class name would apply a 16.66% width to it; ie: one div is used, not two. Each div in a row then has a float:left applied to them to align them in the same row next to each other. Essentially it's divs floated against each other with percentage widths applied to them that all total to 100%, nothing more than that. This is almost the identical layout that we began using all the way back in November 2009 when we switched from a pixel based system to percentage based, so we are very familiar with it. The term Bootstrap uses is called "scaffolding" and you can read more about it here:

http://twitter.github.com/bootstrap/scaffolding.html

The goal here is to make the frontend of templates responsive to allow the sites to fit to any screen size, which Vertex templates have been doing since June of this year. This is all done through percentage based layouts and media queries to change the layout. Bootstrap's responsive layout is simply an alternative for template developers who have been building their templates on a pixel grid based system, and are being forced to change due to the recent demand of responsive layouts. Because of the percentage based layout we have been using for years the transition to responsive layouts was very easy to make for us, and it did not require us to start over from scratch with an entirely new layout; rather we simply added the needed media queries to change our already percentage based layout into a responsive layout. That being said we have looked at the Bootstrap layout quite extensively, and we do agree that is a very good layout and they have done a remarkable job with it, but the layout is almost identical to the approach that we are currently using in Vertex, and in some ways it would actually be a downgrade to our current framework if we replaced our Vertex layout with Bootstrap. The additional features that Vertex includes are detailed below:

1. Custom Row Sizes - This is one of the top features that we offer in Vertex. In any given row of modules in a Vertex template you can specify the exact width of every module in that row. This is all done very easily in the Vertex admin. In boostrap you are simply stuck with the pre-determined widths that are set by the class name of that particular module, ie: the 12 sections/columns in each row that are described above. For example if you wanted a module row to have three modules at 37%, 22%, and 41% you could do that very easily in Vertex. In Bootstrap you can't get widths that specific, it simply goes by how many sections/columns you want that particular div to stretch.

2. Fixed or Fluid - Our layout has the option for a fixed or fluid layout, both which will work with responsive. In addition to this there is a maxium body width, which does not allow your site to get any larger than the width that you determine. The maximum body width feature is useful to prevent your site from becoming too large on very large screens.

3. Optional Responsive Layout - In Vertex you have the option to disable the responsive layout and use a standard layout instead. We must admit that this feature will probably grow less popular as time goes on, but it does allow developers the flexibility to design the site how they want, and not be forced into a particular layout.

4. Column Width Changes - In boostrap the left and right columns are percentage based, this can cause the columns to be extremely wide on large monitors and out of proportion, and very narrow on smaller screens. In a Vertex template they are set to a pixel width that you determine so you always know what size the columns will appear at. If you wish you can enable a feature that will increase the columns on larger screens and reduce them on smaller screens as well. This feature gives the columns a much more consistent look on every screen.

5. Row Module Width Redistribution - If you are using a lot of modules in one particular row this can cause your content to look squished when viewing your site on a narrow screen. The row module redistribution feature allows you to split a single row of modules into several rows to give each module more room. You get to control how how many modules show on each row, and the exact screen pixel width that the new layout will trigger at. Each row can be individually configured. Bootstrap does not have this feature and other template providers have compensated for this by hard coding specific widths into the css files rather than allowing you to choose.

6. Desktop Link - Even though Vertex uses a fully responsive layout and there is no separate layout page for mobile devices, that does not mean that your site's viewers will understand this or even know what a responsive layout is. It has become a standard for websites that use a mobile layout to have a link to view the desktop view of the website, and this is what your site viewers will expect to see. This link simply turns off the responsive configuration using a cookie and calls the website like a standard desktop would when viewing in mobile. There is then a link to return back to the mobile view of the site.

7. Hide Classes - Vertex includes several dozen hide and show classes that allow you to hide or show content or

modules at specific screen widths. This is used for content or extensions that do not work well at specific screen widths. Bootstrap does contain this feature but by default there are only three hide classes and three show classes available to work with. In addition to hide classes you can also hide entire sections of the template directly through the Vertex admin configuration's user interface by selecting areas from a list.

8. Mobile Size - One thing that we do not like about the Bootstrap layout is how early the mobile layout triggers. The mobile/single column, layout is triggered at 767px in Bootstrap. This essentially means that all 7 inch tablets and below will be forced to view the single column layout. It is our opinion that this is too early. The Vertex layout does not trigger a single column layout until 580px, which is the width of most larger cell phones. You can always use the Row Module Width Redistribution if you wish to change a particular row to single column or two columns if need be for smaller screens.

9. Column Equalizer - This is a javascript feature built into Vertex that sets the height of each module in a row to the same height, instead of just allowing each module to stretch to their own height based on their internal content.

As stated above we will be including the Bootstrap js and css file calls for extensions, and future elements of our templates, but as for the layout itself we will be sticking with our Vertex responsive layout because of the benefits mentioned above. Going with this approach will not cause any issues with default Joomla or third party extensions. The features mentioned above will give Vertex templates many more features than a standard Bootstrap template, and ultimately much more flexibility for users and developers.