

# Tabbed Brochure Template Version 4.0 Documentation

PLEASE NOTE: This solution requires the use of javascript in the text interface. The use of scripts in the text interface is neither tested nor supported functionality in the software. This means that the Terra Dotta support team will not be able to provide assistance in the implementation or troubleshooting of this script, and although this functions in the current version of the software, there is no guarantee that a script will continue to function as expected in future versions. Additional support could result in professional service fees.

For complete information regarding the use of javascript to make adjustments to the interface, the following webinar will provide details about the security, implementation, and maintenance of scripts in the text interface:

<https://tdsupport.force.com/support/articles/General/WEBINAR-A-Fistful-of-Javascript>

## Overview

This documentation shows how to implement a tabbed interface on the program brochure page using javascript and CSS to manipulate the content of the brochure. This is done by having one or more pre-built templates in the site's document center that can be fetched through the program builder by selecting it from a drop-down menu. This template has the proper markup and attributes that a script on the public brochure page will transform into tabbed sections. The public brochure page also will have an embedded style tag that determines the appearance of the tabs.

In the WYSIWYG Editor on the 'Brochure' tab of the program builder, the content is displayed in sections based on a template deployed to the WYSIWYG Editor by a button. When this content is viewed on the public website, the contents of each of the sections are turned into tabs that can be clicked on by the user to switch between them.

Implementation of this involves the following steps:

Adding the script/CSS tags to the public brochure page interface

Creating the template with the desired tab names and placeholder content

Uploading that template to the document center

Uploading the template importing script to the document center

Adding the script for importing the template into the program builder's text interface

Once this has been done, the new options will be available in the program builder and can be used to import the template. When importing a template, existing brochure content that is not in the 'tabbed' format will be put into the first tab of the template.

## What's new in version 4.0

- Brochure content conversion from 3.0 > 4.0 button
- Responsive design drop-down menu for smaller screens
- Change from div to section tags for content areas to address issues with lists

## Code files

There are [five code files](#) that are used to implement this tabbed brochure template. [These files](#) are zipped together and are stored alongside these instructions in the "Knowledge" area of the Terra Dotta support portal. The files are:

TabbedBrochureScriptCSS.html - The style and script embedded on the public brochure page

TabbedBrochureTemplate.html - The html that is put into the document center as the template

TabbedBrochureTemplateBuilder.html - A standalone html page that can be used to generate the html of a brochure template

TabbedBrochureTemplateImport.html - The script that loads the template functions and builds the interface in the program builder

TabbedBrochureTemplateImportScript.js - The javascript with all the functions that the interfaces use to manage the templates

## Tab styles

Before implementing the script, you may edit the style of the tabs that will be displayed.

The appearance of the tabs on the public brochure pages can be adjusted by manipulating the 'style' tag of the code file 'TabbedBrochureScriptCSS.html'. To edit the style, open the file 'TabbedBrochureScriptCSS.html' in a plain text editor and use the information below to make changes as you see fit.

You can also use the 'TabbedBrochureTemplateBuilder.html' to manipulate the CSS and see the changes in real time. It is a standalone web page that has an editable style block that is used to display the template tabs being created.

This directive affects the margin and padding around the tabs:

```
#brochure-tabs-navigation ul {  
  list-style: none;  
  padding: 0px;  
  margin: 1px 0px 15px 0px;  
}
```

This directive affects all the tabs and sets the base style of them:

```
.section-tab {  
  border-style: solid;  
  border-width: 0px 2px 2px 2px;  
  display: inline;  
  padding: 3px 10px;  
  margin: 5px;  
  border-radius: 0px 0px 10px 10px;  
  font-family: Arial;  
  font-weight: bold;  
  background-color: #FFFFFF;  
  color: black;  
  border-color: black;  
  cursor: pointer;  
}
```

This directive affects the style of a tab when the user is hovering the mouse cursor over it:

```
.section-tab:hover {  
  box-shadow: 0px 1px 10px -1px #000000;  
  background-color: orangered;  
}
```

This directive affects the tab which is the currently displayed tab:

```
.selected-section-tab {  
    box-shadow: 0px 1px 10px -1px #000000;  
    background-color: powderblue;  
}
```

## Implementing the script/CSS

Once the desired changes have been made to the style directives in 'TabbedBrochureScriptCSS.html', it is ready to be appended to text interface field# 3 (program brochure page tips text).

This can be done before the template is built and implemented. The script and style tags will only impact the page if there is a valid template in the contents of brochure so putting it in place as soon as the tab styles are done is fine. It will not adjust pre-existing brochure page content.

## Creating the templates

The templates are HTML that will be put into document center WYSIWYG objects. This HTML markup will be fetched and inserted into the WYSIWYG Editor of the program builder's brochure editor so that a valid template with the necessary markup and attributes can be easily deployed.

Before creating templates, the following will need to be decided:

1. How many templates will you have and what will you call them
2. What will the tabbed sections be called in each template
3. What order should the tabs appear in
4. What placeholder text should be in each section in the template

There are two ways to create the template. The easier way is to open the file 'TabbedBrochureTemplateBuilder.html' in your browser. This is a standalone webpage that can be used to build a template that will auto-generate the html you can use as a template.

Alternatively, you can manually build templates yourself. The file 'TabbedBrochureTemplate.html' has a valid template with three tabs called 'Tab1', 'Tab2', and

'Tab3'. To make changes to this template before you implement it, open the file 'TabbedBrochureTemplate.html' in a plain text editor. You will see the same code that you see here. You can edit the template based on the information below, as you see fit.

```
<!-- Upload to document center as WYS object -->
<div id="brochure-tabs-content">
  <div class="tab-header">
    <h1>Tab1</h1>
  </div>
  <section class="tab-content" id="Tab1">
    Lorem ipsum
  </section>
  <div class="tab-float-clear" style="display: table; clear:
both;">&nbsp;</div>
  <div class="tab-header">
    <h1>Tab2</h1>
  </div>
  <section class="tab-content" id="Tab2">
    Lorem ipsum
  </section>
  <div class="tab-float-clear" style="display: table; clear:
both;">&nbsp;</div>
  <div class="tab-header">
    <h1>Tab3</h1>
  </div>
  <section class="tab-content" id="Tab3">
    Lorem ipsum
  </section>
  <div class="tab-float-clear" style="display: table; clear:
both;">&nbsp;</div>
</div>
```

The div tag after the content div is there to make sure that the sections don't wrap around each other. It makes the brochures more easily edited in the program builder and display in printed versions more cleanly. That div should not be changed.

To change one of the tabs, the content wrapped in 'h1' tags of the first 'div' and the 'id' of the next 'section' should be edited as you see fit. The text between the 'h1' tags is the Header that will appear in the WYSIWYG Editor of the program builder when that brochure is being edited so that the admin will know which tab's content they are editing. The 'id' is the name of the tab. In other words the 'id' is the word that will appear on the tab itself.

NOTE: The name of the tab ('id' in the section) cannot contain spaces.

The 'Lorem ipsum' in the second div can be adjusted to whatever is desired as placeholder content.

Example - to change the first tab in the template to 'General', you would change:

```
<div class="tab-header">
  <h1>Tab1</h1>
</div>
<div class="tab-content" id="Tab1">
  Lorem ipsum
</div>
<div class="tab-float-clear" style="display: table; clear:
both;">&nbsp;</div>
```

to:

```
<div class="tab-header">
  <h1>General content</h1>
</div>
<div class="tab-content" id="General">
  Lorem ipsum of general section
</div>
<div class="tab-float-clear" style="display: table; clear:
both;">&nbsp;</div>
```

If more than three tabs are needed, you can append additional groups of 'div's to the file.

If a tab doesn't need to be used for a program, the template can be imported into the WYSIWYG Editor and then the placeholder text and header deleted from it. Then, that tab won't appear on the public brochure page.

If a tab needs to be renamed/added for a specific program, there are buttons that can be used to do that in the program builder.

Once the template's tabs have been configured, the HTML file should be saved, and it is ready for implementation.

## Implementing the templates

To implement templates on a site, you first need to upload the file 'TabbedBrochureTemplateImportScript.js' to your document center as a script. This script has the functions needed by the program builder's interfaces to implement the templates.

To do this, following these steps:

- Go to Website Admin -> Document Center
- Click on 'New Document'
- Under 'Option 2' click the browse button and select 'TabbedBrochureTemplateImportScript.js' off your local drive
- Give the document a name (the title doesn't matter)
- Click the 'Save' button

NOTE: It is possible that your site is not configured to allow .js files to be uploaded to the document center. If you have access to Maintenance > Environment Settings, you can temporarily allow that file extension to be uploaded. Alternatively, you can rename the file so that it has a .txt file extension rather than .js. This will still work in the program builder.

Next, the templates need to be uploaded to the document center as a WYSIWYG Objects. To do this:

- Go to Website Admin -> Document Center
- Click on 'New Document'
- Enter a title for the template for 'Document Title' (the title doesn't matter)
- Click on the 'Source' button in the WYSIWYG Editor
- Copy/Paste the HTML of the template into the WYSIWYG Editor
- Click 'Save' button

After saving the templates to the document center, it is necessary to get the 'File\_ID' of each template. You can find these number by clicking on the 'view' link of the templates and looking in the URL of the resulting page.

These numbers will be used to create hidden input tags in the script in the file 'TabbedBrochureTemplateImport.html' that identifies what the template should be called in the drop-down menu around line 24:

```
<input class="template-name-id" value="{TEMPLATE NAME}|{FILE ID#}"
/>
<script>
    $(document).ready(function() {
        $.getScript('{URL OF TEMPLATE SCRIPT IN DOCUMENT CENTER}',
function () {
    initTabs();
});
```

```
});  
</script>
```

For each template, there will need to be an input tag using that format to identify what to call. So, for example, if I had two templates to implement; one that was the standard and another for provider programs, the resulting block of html might look like this:

```
<input class="template-name-id" value="Standard|31" />  
<input class="template-name-id" value="Provider program|32" />  
<script>  
    $(document).ready(function() {  
        $.getScript('{URL OF TEMPLATE SCRIPT IN DOCUMENT CENTER}'  
function () {  
        initTabs();  
    });  
});  
</script>
```

In addition to the input tags, it is necessary to put in the document center URL to the import interface's script into the {URL OF TEMPLATE SCRIPT IN DOCUMENT CENTER} placeholder. The 'http:' or 'https:' of the URL should be removed to avoid cross-protocol issues. So, for example, if my site's domain were <http://studyabroad.terradotta.com> and the file ID# of the script was 33, the resulting code would look like this:

```
<input class="template-name-id" value="Standard|31" />  
<input class="template-name-id" value="Provider program|32" />  
<script>  
    $(document).ready(function() {  
  
$.getScript('//studyabroad.terradotta.com/_customtags/ct_FileRetrie  
e.cfm?File_ID=33' function () {  
        initTabs();  
    });  
});  
</script>
```

This script should be appended to the text interface field #4400. This will create the button in the program builder that fetches the brochure template from the document center. It will also add buttons for adding new tabs in the brochure and deleting existing tabs in the brochure.

Once this has been done (and the file 'TabbedBrochureScriptCSS.html' has been implemented) , the template can be used by anyone by selecting the desired template in the drop-down menu



and then modifying the placeholder text in the WYSIWYG Editor. Each section will be turned into a separate tab when viewed on the public website.

## Adding tabs to a brochure

NOTE: While the import of a template will take the current content of the brochure and put it into the first tab of the template. It is recommended to make a backup of brochure content you are converting over to the tabbed-brochure content. Create a copy of your original brochure source in a plain text file. By following these steps.

1. Navigate to the brochure tab of the program builder for the program you wish to edit.
2. Enter 'source' mode in the WYSIWYG editor.
3. Select all of the code in the WYSIWYG editor and copy it.
4. Paste the code into a plain text editor document. Save the document with a name that indicated the program from which it came.

You may also want to take a screen capture or print a copy of the brochure you are about to edit.

Now you are ready to create a brochure page with tabs. On the brochure tab of the program you wish to edit, select the brochure template from the drop-down menu and confirm you want to import the template. If the current content of the WYSIWYG Editor is 'non-tabbed brochure' content, it will put that content in the first tab of your template. You can now edit the brochure within the WYSIWYG editor in WYSIWYG mode. Edit each section. You can edit the header of each section as well as the body text. You can also insert images, and you should be able to use all of the functions available in the WYSIWYG. When you update the brochure, your changes will be saved immediately and you can preview your page by navigating to the public interface.

There are also additional buttons that can be used to add and remove tabs from the brochure you are currently editing.

Finally, if you used a Version 3.0 tabbed brochure template previously for tabbed brochure content, the structure of the templates changed a little bit in this version to address issues with lists breaking the tab markup. There is a convert button that can be used to update the current contents of the WYSIWYG Editor from the 3.0 template markup to the 4.0 template markup.

## Upgrading from Version 3.0 to 4.0

To update the Version 3.0 implementation of the tabbed brochure to Version 4.0, the following must be done.

1. Update the existing brochure templates in the document center
2. Update the TabbedBrochureTemplatelImport.html content in the program builder
3. Update the TabbedBrochureTemplatelImportScript.js script in the document center
4. Update the TabbedBrochureScriptCSS.html content on the program brochure page
5. Update program brochure content using the conversion button

## Update the existing brochure templates in the document center

The markup of the brochure templates has a minor change between 3.0 and 4.0. Instead of using a 'div' tag for each of the content sections, a 'section' tag is used. This improves the semantic nature of the template, but more importantly, it solves a few issues that had been reported with the WYSIWYG Editor when trying to include ordered and unordered lists within the brochure pages.

To update your templates, you just need to change the content sections of your template that look like this ...

```
<div class="tab-content" id="Tab1">
  Lorem ipsum
</div>
```

... to use a section tag so that they look like this ...

```
<section class="tab-content" id="Tab1">
  Lorem ipsum
</section>
```

Nothing else in the template will need to change. The surrounding markup and headings can remain unchanged.

## Update the TabbedBrochureTemplatelImport.html content in the program builder

The markup that has been embedded into the program builder's text interface to allow importing the templates (TabbedBrochureTemplateImport.html) has been updated and will require you to update the code.

First, go to Maintenance > Text interface and make a copy of the contents of text interface field #4400 - save this content locally in a text file.

Next, open the new 4.0 version of the file TabbedBrochureTemplateImport.html in a text editor.

Replace this line of the new version with the input tags that you used in the 3.0 version ...

```
<input class="template-name-id" value="{TEMPLATE NAME}|{FILE ID#}"
/>
```

Then, replace this line of the new version with the URL that you used in the 3.0 version ...

```
$.getScript('{URL OF TEMPLATE SCRIPT IN DOCUMENT CENTER}', function
() {
```

This updated version of the file should replace the old version of the markup in text interface field #4400.

## Update the TabbedBrochureTemplateImportScript.js script in the document center

There is no need to make any updates to the new version of the file. It should just replace the existing TabbedBrochureTemplateImportScript.js script you have in your document center. You should be able to just go to Site > Document Center and click the edit pencil for the script. The new version can be uploaded in its place.

## Update the TabbedBrochureScriptCSS.html content on the program brochure page

While your current brochure tabs will work with the Version 3.0 script and style block that is in text interface field #3, if you want to implement the responsive version of the tabs, you will need to update the content.

First, go to Maintenance > Text interface and make a copy of the contents of text interface field #3 - save this content locally in a text file. Add the following directive to the contents of the style block ...

```
.tab-float-clear {  
    display: none !important;  
}
```

Next, open the new 4.0 version of the file TabbedBrochureScriptCSS.html in a text editor. Copy the contents of the style block from your old version of the file into the style block of the 4.0 version of the file.

This updated version of the markup should replace the old version of the markup in text interface field #3.

## Update program brochure content using the conversion button

Finally, the last step is to go to the 'brochure' tab in the program builder for all your programs that were using the Version 3.0 template and click on the 'Convert template' button. This will update the content of the brochure to make it work with the Version 4.0 scripts on the brochure page (changing the 'div' tags of your content sections to 'section' tags).

## Technical details

The tabbed brochure template uses the ability to embed scripts and styles in the text interface to manipulate the contents of the public program brochures.

There are five files involved to implement this system on a site:

### TabbedBrochureTemplate.html

This is an example template that can be turned into a site's template. It is where the names of the tabs are determined and the order in which they are going to be displayed.

There is a container 'div' with the id 'brochure-tabs-content' wrapping all the brochure's content so that it can be more effectively targeted by the script on the brochure page.

Each tab in the template has a 'div' and a subsequent 'section'. The div has the class 'tab-header'. The content of this tag is not displayed on the public brochure page. It is only displayed in the program builder's WYSIWYG Editor so that the admin knows which tab the subsequent content will be in. The section has two attributes; the class 'tab-content' and then an id which will be the name of the tab on the public brochure page. The content of this section is what will be displayed on the public brochure page for that tab and can have whatever placeholder content is desired by the admin setting up the template.

There can be as many tabs as needed, but the names of the tabs must be valid ids which means that they cannot contain spaces. If a space is needed,  can be used.

There is also a div tag after each content section that makes it easier to edit and print pages that float content left or right. It clears any floats between the sections.

## TabbedBrochureTemplateBuilder.html

This is a lightweight html page with javascript that allows the user to create the tabbed brochure template in a UI rather than by manually hand-coding the template's html. All the user has to do is drag the html file into their browser to open it, and it will present an interface that can be used to build the tabs needed and generate the html that can be copy/pasted into the WYSIWYG Editor of the document center.

## TabbedBrochureTemplateImport.html

This markup gets embedded into the text interface of the program builder's 'brochure' tab. It adds the interface elements that are used to fetch the tabbed brochure template from the document center and put it into the WYSIWYG Editor. It also adds buttons that allow the user to create new tabs, remove existing ones, and update the markup to the latest version of the brochure tabs in the brochure they are working in. It also loads a script from the document center that has all the functions that are driven by the interface.

## TabbedBrochureTemplateImportScript.js

This is the script that has all the functions that are used by the interface to import and modify the tabs in the WYSIWYG Editor.

## TabbedBrochureScriptCSS.html

This file has the script and style tags that get embedded into the text interface of the public program brochure page and transform the program brochure contents into a tabbed interface if they have proper markup in them.

The style tags define the way that the tabs should look. There is a class for all the tabs, a class for when the user is hovering over a tab, and a class that also gets assigned to whichever tab is currently being displayed on the page.

There is also a directive for the entire navigation list which can be used to adjust the surrounding margins and padding.

The script has a number of commands that run on the document's ready event. It builds an array of the ids of any 'tab-content' classed elements in the 'brochure-tabs-content' id container.

It then builds a div with an unordered list that is only displayed on larger screens containing each of the tabs as well as drop-down menu that is only displayed on smaller screens. This div gets prepended to the top of the brochure's content where the styles make the list items appear as tabs.

Then, the script hides all the content of the brochure and reveals only the first tab's content. It also assigns a click event handler to each tab. The function 'showBrochurePanel' is used as this click event handler to first hide all the contents of the brochure and then reveal only the contents of the tab that was clicked.

This tabbed brochure template has been tested in TDS 16.0 in Chrome Version 51.0.2704.106 (64-bit) and Firefox Version 45.0.1.