

Creating a custom theme

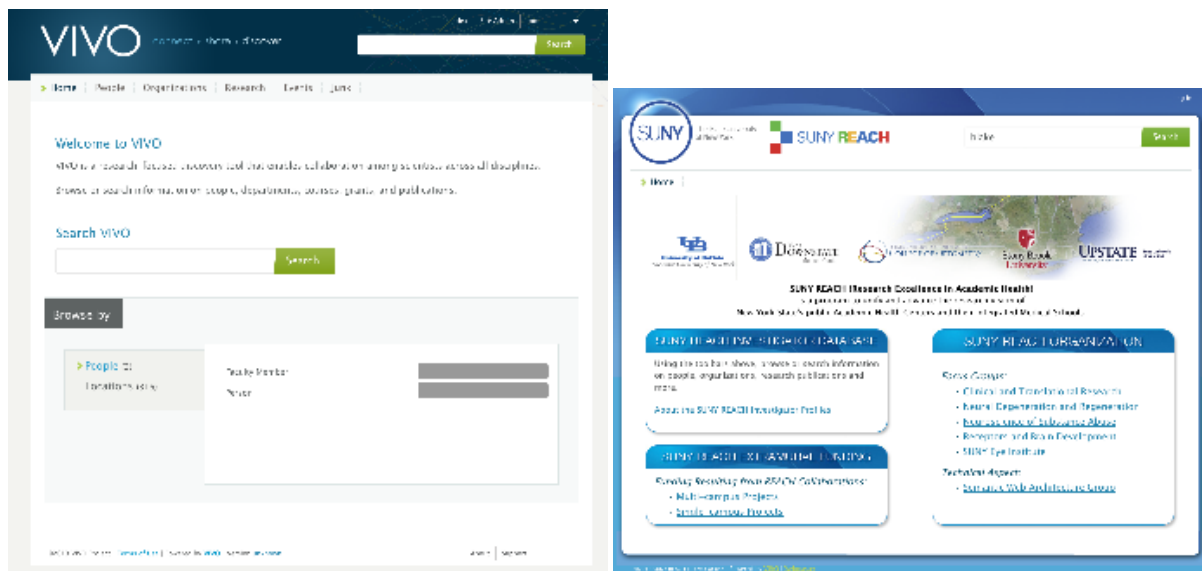
- Overview
 - What can it do for you?
 - Before and After
 - What do you need to know?
 - Getting started
- The structure of pages in VIVO
- Some significant templates
- Making changes
 - Modify files in the theme
 - Add files to the theme
 - Add CSS, JavaScript, or image files
 - Add Freemarker templates
 - Override files that are not in the theme directory
 - Override CSS, JavaScript or image files that are not in the theme directory
 - Override Freemarker templates that are not in the theme directory
 - Working on the theme
 - When to restart Tomcat

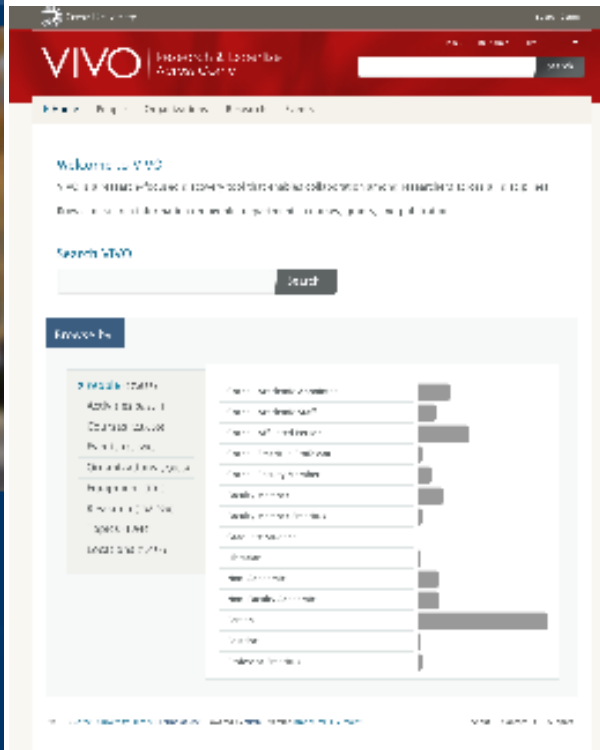
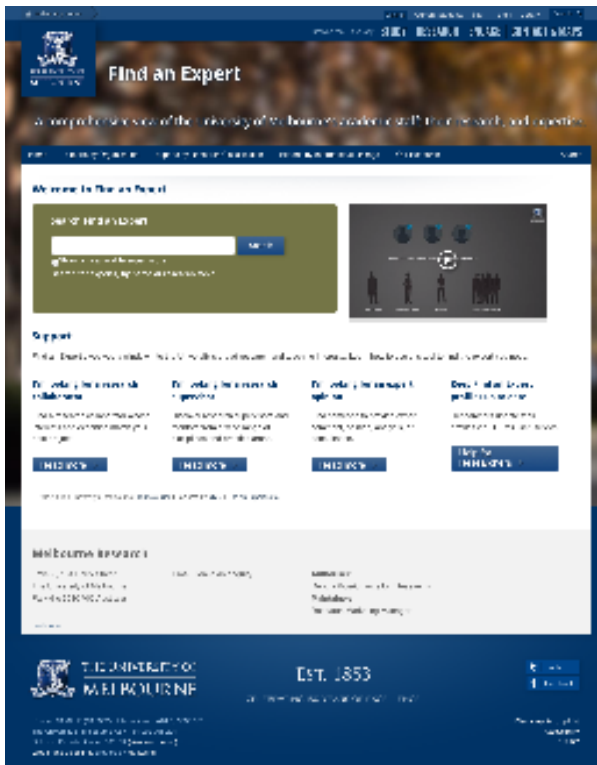
Overview

What can it do for you?

Change the "look and feel" of your VIVO installation. Change the styling, the images, the layout, the text, and more. Modify the header and footer on all pages.

Before and After





What do you need to know?

- Standard web-site technologies: HTML, CSS and maybe JavaScript.
- Something about the [Freemarker](#) template engine.
- Where the theme files are stored in VIVO, and how to reference them.

Getting started

VIVO comes with a standard theme, called `wilma`. `wilma` is in the folder in `vivo/installer/webapp/target/vivo/themes`.

To create a new theme, choose a name for your new theme. In these examples below we will call the new theme `fred`.

Copy the `wilma` directory and its contents to a new directory called `fred`. `fred` must also be in `vivo/installer/webapp/target/vivo/themes`.

Your new theme will contain CSS files, image files, and [Freemarker](#) templates.

Run the Maven install to deploy your new theme to the Tomcat container. Restart the VIVO Tomcat process. You can then go to the **Site Admin** page and choose **Site Information**, to select your theme as the current one.

Site Information

Editing Existing Record

Site name (max 50 characters)

Contact email address contact form submissions will be sent to this address

Theme fred ▾

Copyright text used in footer (e.g., name of your institution)

Copyright URL copyright text links to this URL

Save changes

Cancel

The structure of pages in VIVO

The pages in VIVO are built around three different frameworks. Each of these uses the same header and footer, to provide consistency. In addition to including the header and footer, the pages frequently include smaller templates to provide detail.

These are the basic frameworks:

The home page	As the point of entry for VIVO, the home page is special. It is based on the Freemarker template <code>page-home.ftl</code>
All other public pages	Based on the Freemarker template <code>page.ftl</code>
"back-end" pages	Pages used for editing the ontology, or manipulating the raw data of VIVO are based on a JSP named <code>basicPage.jsp</code>

Some significant templates

page.ftl

`page.ftl` is the default base template. The rest of the theme templates listed are components of `page.ftl` (included either directly or indirectly). Closer inspection of `page.ftl` reveals a stripped down file that declares minimal markup itself and instead reads as a list of includes for the component templates.

On the VIVO home page, `page-home.ftl` is used instead of `page.ftl`. It serves much the same purpose, but allows you to create a different layout for your home page than for the other pages in VIVO.

For consistency, it is critical that the following components be maintained:

head.ftl

This component template is responsible for everything within the `<head>` element. Note that the open and closing tags for the `<head>` element are defined in `page.ftl` and wrap the include for `head.ftl`. There are several includes within `head.ftl` that should be carried over to any new theme to maintain expected functionality:

- `<#include "stylesheets.ftl">` - ensures that the necessary stylesheets called by templates downstream will be added to the page via `<link>` elements
- `<#include "headscripts.ftl">` - ensures that the scripts called by templates which must be in the `<head>` will be added to the page via `<script>` elements

identity.ftl

This component template is responsible for rendering the VIVO logo, secondary navigation and search input field at the top of the page. There are no mandatory includes from `identity.ftl` that need to be carried over but there are 2 template variables that are of particular interest (`${user}` and `${urls}`).

menu.ftl

This component template is responsible for rendering the primary navigational menu for the site. In Wilma, it also happens to declare the open tag for the main content container. There are no mandatory includes from `menu.ftl`. The `${menu}` template variable is crucial since it contains an array of menu items needed to build the primary navigational menu.

footer.ftl

This component template is responsible for rendering the copyright notice, revision information, secondary navigation, and link for the contact form. There is a single include that should be maintained:

- `<#include "scripts.ftl">` - ensures that the non head scripts (those that don't need to be placed in the `<head>`) called by the templates will be added to the page via `<script>` elements

Several template variables of interest include `${copyright}`, `${user}`, and `${version}`.

googleAnalytics.ftl

This component template is included by `footer.ftl`. Simply uncomment the `<script>` element and provide your Google Analytics Tracking Code.

Adjust the markup as necessary in `page.ftl`, and these component templates to achieve the desired content structure, and modify the stylesheets to meet layout needs and style your site. Remember that changes should be made in the source directory and that you will need to redeploy the project before the changes are reflected in the live website.

You can find more information about the structure of the VIVO theme in [How VIVO creates a page](#).

Making changes

Modify files in the theme

You can edit the Freemarker templates and the CSS files in the theme with any text editor. You can replace the image files with images that you choose.

Add files to the theme

Add CSS, JavaScript, or image files

As you modify the templates, you may want to use additional images, CSS files, or JavaScript files. When your templates refer to these files, they will use the Freemarker variable `urls.theme`, as shown in these examples:

```
<!-- an image file -->


<!-- a CSS file -->
<link rel="stylesheet" href="${urls.theme}/css/screen.css" />

<!-- a JavaScript file (create a js directory in your theme) -->
<script type="text/javascript" src="${urls.theme}/js/my.js"></script>
```

Add Freemarker templates

If your modifications use new Freemarker templates, you can refer to them more simply. Freemarker already knows where your theme directory is located.

```
<#include "my-new-template.ftl">
```

Override files that are not in the theme directory

In order to keep the theme directory uncluttered, VIVO keeps most of the front-end files in a separate location. Changes to the theme usually involve the files in the theme directory, but you can override other files as well.

Override CSS, JavaScript or image files that are not in the theme directory

You may notice that templates refer to files that are not in the theme directory. They use references based on the Freemarker variable `urls.base` instead of `urls.theme`, like this:

```
<!-- an image file -->


<!-- a CSS file -->
<link rel="stylesheet" href="{urls.base}/css/login.css" />

<!-- a JavaScript file -->
<script type="text/javascript" src="{urls.base}/js/browserUtils.js"></script>
```

These refer to files in the `vivo/installer/webapp/target/vivo` directory. If you look, you will see that this directory contains some files also used in the construction of the VIVO interface.

Override Freemarker templates that are not in the theme directory

To override templates not in the theme directory, simply modify Freemarker templates in `vivo/installer/webapp/target/vivo`. These changes will apply to all your themes.

VIVO treats all available Freemarker templates as belonging to the same flat namespace, whether they are in the theme directory or in the `templates/freemarker` directory, or one of its sub-directories. A file in `vivo/installer/webapp/target/vivo` can be overridden by a corresponding file in the theme directory.

Working on the theme

When you make changes to VIVO, you should make the changes in your VIVO distribution directory, run Maven install, restart Tomcat, and test the changes. If you are doing full customizing of VIVO, this cycle might be best.

If you are only working on the theme, you can speed things up.

- Tell the build script to skip the unit tests: they don't test the theme
 - `mvn install -DskipTests=true`
- Don't restart Tomcat
 - VIVO always serves the most recent version of CSS files, image files, and JavaScript files. You don't need to restart Tomcat to make that happen.
 - However, your browser may cache these files so you won't see the most recent version. Here are some suggestions for [bypassing your browser cache](#).
- Tell VIVO to reload Freemarker templates each time they are requested. See [Tips for Interface Developers](#).

Some developers prefer to make theme changes inside the `tomcat/webapp/vivo` directory. This eliminates the need to run the build script, but opens the threat of having the changes over-written the next time the build script runs.

When to restart Tomcat

If you make changes to any of the source files in the theme, including images, CSS, JavaScript or Freemarker templates, you must run the build script, but you do not need to restart Tomcat.