Upgrading VIVO

Upgrading from 1.8.x to 1.9.x

There are no ontology changes, and no significant changes to the code. However, there are a few small changes that you will need to make to your configuration in order for everything to work - in particular, see the **web.xml** change.

You will also need to move your customizations into the new Maven structure.

Maven Structure

In moving to Maven, the code has now been re-organized to follow standard Maven conventions. As well as separating out the Java code into "API" projects for both Vitro and VIVO, with the templates, etc. in "webapp" projects, it also means that the directory structure follows the standard. So in "API", all the Java code is in the "src/main/java" directory, and in "webapp" the templates are in "src/main/webapp".

It is recommended that you treat the "installer" (or standard distribution) project(s) as a third tier, and keep local customizations only in these project(s).

For Java files, you can place them directly in the "webapp/src/main/java" directory - these will be compiled and included into your web application as class files. As class files are loaded ahead of jars in WEB-INF/lib, this even allows you to completely overwrite and replace classes that are supplied by the Vitro / VIVO jars.

Your theme - and any other template modifications, etc. - should be placed in the "webapp/src/main/webapp" directory.

Any additional dependencies that you need can be added to the webapp/pom.xml, and they will be retrieved from Maven central and included in the packaged application.

Making changes to the home directory

In order to add files to your home directory - e.g. to include additional filegraph RDF - they should be added to the "home/src/main/resources" directory.

If you need to remove a file that is supplied by Vitro / VIVO, then you will need to edit the home/src/main/assembly/home.xml file:

```
<dependencySet>
    <outputDirectory>/</outputDirectory>
    <unpack>true</unpack>
    <unpackOptions>
        <excludes>
            <exclude></exclude>
        </excludes>
    </unpackOptions>
</dependencySet>
```

Code / Environment Changes

Capability Map

The Capability Map is only linked from the main menu bar, which is dynamically generated from statements in the triple store. The file that includes these definitions is only loaded once during the first startup of VIVO.

In order to add a menu entry to an instance during an upgrade, the easiest way is to add a page via the administration functions.

First, go to "Site Administration" (e.g. after logging in as an administrator, the "Site Admin" link at the top), and then choose "Page Management", and click on the "Add Page" button.

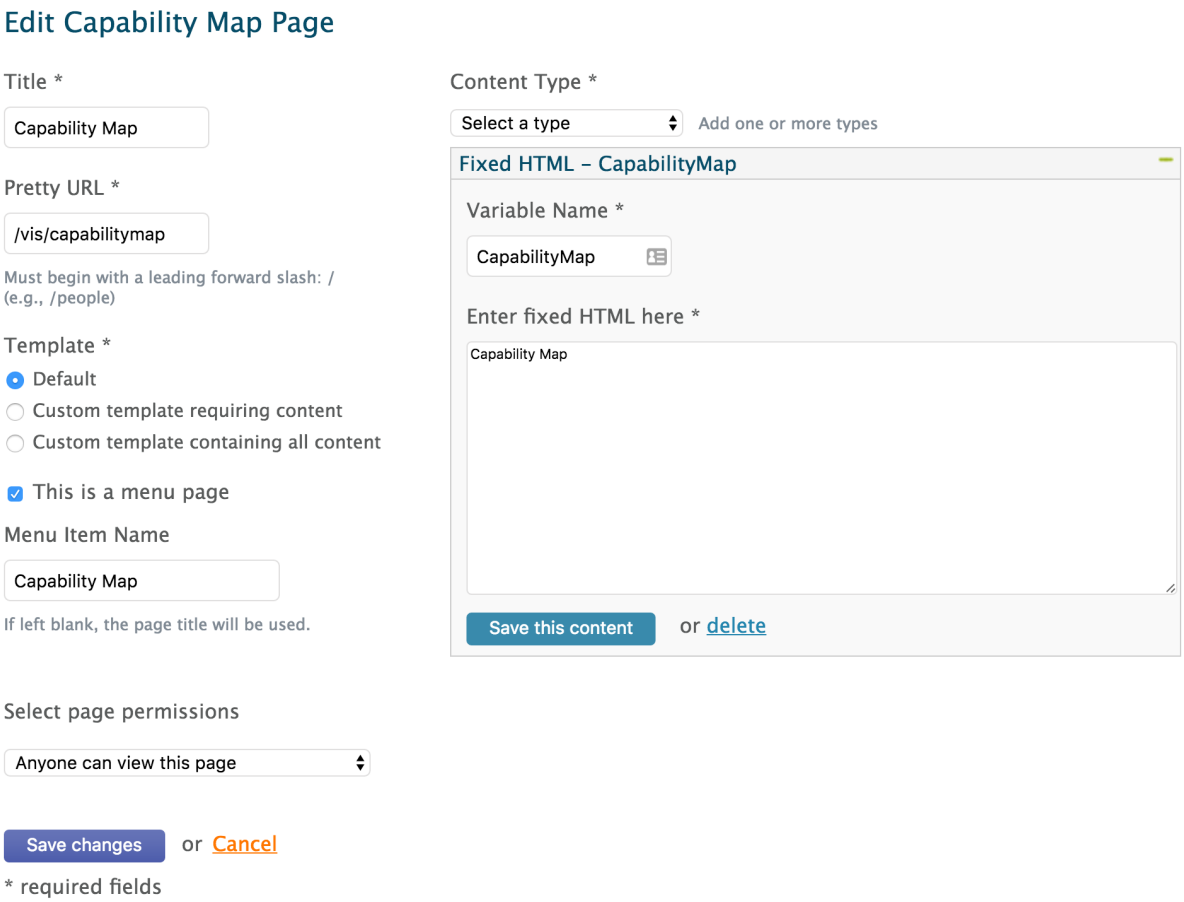
Provide a title, select "This is a menu page", and give a name to the menu entry.

For "Pretty URL", you must enter the following:

```
/vis/capabilitymap
```

You will need to choose a content type - select "Fixed HTML" and then enter anything you like in the fields as they won't be displayed.

A complete configuration should look like this.



Once saved, you should see the new entry in the menu bar and be able to navigate to the capability map.

The Capability Map depends on having People with Research Areas defined. If you do not have enough research areas to make the capability map useful, you can remove the page from the menu via Page Management.

AltMetrics on Profiles

In order for AltMetric badges to be displayed in publication lists in profiles, the AltMetric script needs to be included in the page. This would usually be in the individual--foaf-person.ftl, which is included in the "wilma" theme.

If you are using your own theme, you will need to ensure the following is part of your theme's templates/individual--foaf-person.ftl:

```
${scripts.add('<script type="text/javascript" src="https://d1bxh8uas1mnw7.cloudfront.net/assets/embed.js"><
/script>')}
```

Map of Science

You should obtain a developer key from Google to use the Maps API. Please follow the guide here: https://developers.google.com/maps/documentation/javascript/get-api-key.

When you have a key, you will need to add it to your runtime.properties:

```
google.maps.key=<insert your key>
```

Freemarker 2.3.23

The Freemarker library has been updated to 2.3.23. The only known code conflict - edu.cornell.mannlib.vitro.webapp.freemarker.config.FreemarkerConfigurationImpl.java - has been updated to have the correct method signature for getTemplate(). This is unlikely to affect any custom templates, however you should make a note to check them.

web.xml

There are two important additions and one change that have been made to the web.xml file. If you are using a custom web.xml file, you need to ensure that you replicate them.

ImageProcessor Filter

In order for the new OpenJDK compatible ImageProcessor to function, you need to have the listener configured. Please add the following just after the StartupManager listener:

```
<!-- TwelveMonkeys ImageIO listener -->
<listener>
   <display-name>ImageIO service provider loader/unloader</display-name>
   <listener-class>com.twelvemonkeys.servlet.image.IIOProviderContextListener</listener-class>
</listener>
```

SiteAdminController

The relationship between VIVO and Vitro code has been cleaned up, such that there are now no files in VIVO that entirely overwrite and replace a class of the same name in Vitro. However, in doing this, one VIVO specific servlet was introduced, requiring that the configuration in web.xml is updated.

Replace the line:

```
<servlet-class>edu.cornell.mannlib.vitro.webapp.controller.freemarker.SiteAdminController</servlet-class>
```

with:

```
<servlet-class>edu.cornell.mannlib.vitro.webapp.controller.freemarker.VIVOSiteAdminController</servlet-class>
```

SiteMap and robots.txt

In order to produce the sitemap for profiles, and to embed the correct link to it in the robots.txt, a servlet has been introduced to provide that functionality.

You will need to add the following to a custom web.xml:

```
<servlet>
  <description>SiteMap support</description>
  <servlet-name>SiteMapServlet</servlet-name>
  <servlet-class>org.vivoweb.webapp.sitemap.SiteMapServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SiteMapServlet</servlet-name>
  <url-pattern>/robots.txt</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>SiteMapServlet</servlet-name>
  <url-pattern>/sitemap.xml</url-pattern>
</servlet-mapping>
```

head.ftl and propStatement-*.ftl

To support the output of citation meta tags, a new collection has been introduced called "metaTags". This is output by the theme in the head.ftl template. In your theme, please ensure that the head.ftl includes the following:

```
<#if metaTags??>
    ${metaTags.list()}
</#if>
```

It should appear immediately following the <#include "headScripts.ftl> line.

Also, if you have any custom templates for output property and data statements, please check them against the ones in VIVO that use the lib-meta-tags.ftl import and addCitationTags macro.

applicationSetup.n3

To use the new OpenJDK compatible ImageProcessor, you will need to adjust your applicationSetup.n3.

In the :application section, set the property for :hasImageProcessor as follows:

```
:hasImageProcessor            :iioImageProcessor ;
```

Then, add a definition for :iioImageProcessor:

```
:iioImageProcessor
  a <java:edu.cornell.mannlib.vitro.webapp.imageprocessor.imageio.IIOImageProcessor> ,
    <java:edu.cornell.mannlib.vitro.webapp.modules.imageProcessor.ImageProcessor> .
```