

Theme Tutorial: Getting Started with Manakin Themes

Alexey Maslov, Cody Green, Adam Mikeal, Scott Phillips, John Leggett

This quick guide is intended to walk you through the process of setting up your first Manakin theme. It will take you from the process for creating a blank new theme, configuring that theme, and finally installing it to apply to specific DSpace pages in order to set the stage for further development.

Create a folder where the theme will be stored at in the themes directory.

Locate your themes directory. Depending on whether you work from the Manakin source tree or directly from your Tomcat install, the themes directory will be either directly inside Manakin root or inside the `tomcat/webapps/manakin-dir/`. Create a new directory inside themes and name it after your new theme. Remember the directory name; it will come in useful when configuring the pages that the theme will be applied to.

Create a sitemap file for the theme.

Once a directory is created to house the theme, then next step is to create the theme itself. The heart of a theme is a single Cocoon configuration file called a sitemap that determines how the theme processes the DRI document passed to it from the aspect chain. Through the sitemap the theme can be configured to apply different transformations and add metadata to the DRI document, generate debug output and do various other things. Since creating a new sitemap from scratch requires good working knowledge of Cocoon and Manakin internals, a template sitemap is provided as a starting point. It is located in the “template” directory and is configured with the most common settings as well as the instructions on changing them. Copy the contents of the template directory into your new theme.

Configure the sitemap

A theme’s sitemap is, in general, responsible for performing the following tasks:

1. Generate the DRI page. This step generates a DRI-encoded XML representation of the DSpace page currently being requested.
2. Add page metadata. This step places a reference to the CSS files that the theme will use into the pageMeta section of the DRI document. This allows the XSL that will subsequently convert the DRI document into XHTML to find those references and place them into the head element of the XHTML output.
3. Transform to XHTML. This step references the XSL to be used to convert DRI to XHTML. This could either link to the existing library called `dri2xhtml.xsl` or to the theme’s own XSL code (which may reference `dri2xhtml` itself).
4. Localize the page, calling the Cocoon `i18n` transformer to resolve the `<i18n:text>` tags
5. Serialize to the browser.

The sitemap that comes with the theme template is configured to place references to one or two CSS stylesheets (depending on the browser) into the pageMeta and to use existing XSL library dri2xhtml.xsl to perform the conversion. Generation, localization and serialization are handled as well. In fact, at first, the only change you have to make to the template is to set its new location at the very top of the sitemap file:

```
<map:component-configurations>
  <global-variables>
    <theme-path>YourThemeDir</theme-path>
    <theme-name>The theme's name/description (not required)</theme-name>
  </global-variables>
</map:component-configurations>
```

Your theme is now configured and you are almost ready to get started.

Installing the theme

Before you begin work on CSS or XSL changes, the newly created theme should also be installed. A theme is installed by adding an entry for it in the xmlui.xconf file, found in your DSpace config directory. Locate the <themes> block inside the xmlui.xconf and add an entry to your own theme there. The theme's location is specified by the path attribute while the set of DSpace pages it applies to can be specified in three different ways:

With regex pattern:

```
<theme name=" Theme's name" regex="community-list" path=" YourThemeDir /"/>
```

Directly with a handle:

```
<theme name=" Theme's name" handle="123456789/42" path=" YourThemeDir /"/>
```

Or both:

```
<theme name=" Theme's name" regex="browse-title^" handle="123456789/42"
  path=" YourThemeDir /"/>
```

Do note that only the first theme rule that matches a page request is used. For example, if you place the most general rule (like regex=".*") at the top, it will always be used instead of all others. In addition, theme rules do cascade unless overridden by another theme rule further down. Thus, a theme rule applied to a community will also apply to all of its sub-communities, collections and items. In general, you should place the most specific rules at the top of the themes block and the most general ones on the bottom to take full advantage of the cascading effects. The comments above the themes block in the xmlui.xconf explain the cascading rules further. Finally, Tomcat must be always be restarted for a change in xmlui.xconf to take effect.

Theme development

At this point you are ready to start working on your theme. When editing the CSS, Manakin theme development is much like any Web development project. You start out with XHTML, a blank CSS and possibly a design in mind, and work with the CSS until the results are satisfactory. The only other thing you need to do is to make sure the sitemap references the files you are working with.

When editing XSL, the most common process is to create a blank XSL file in your theme folder, have it import the existing XSL library (dri2xhtml.xsl), and simply override templates as needed. Under the XSL import rules, when one XSL stylesheet imports another, the importing stylesheet gains use of all the imported stylesheet's templates as if they were its own. However, in case of conflict (for example a template in both stylesheets than matches the same element) the importing stylesheet always takes precedence. This allows you to import the existing XSL from your theme's stylesheet, copy the templates you wish to override into your XSL and make your changes. The modified template will take precedence, overriding the existing XSL code.