

## **Manakin Themes: customizing the look-and-feel of DSpace**

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

A cursory examination of the more than 150 registered DSpace instances reveals a striking degree of conformity in style. Although there are many reasons for institutions to customize the look and feel of their repository—such as institutional branding or imparting context—the current JSP paradigm makes this a tedious task. Furthermore, some customization tasks, such as applying a different style to a specific collection or community, are currently not supported. Manakin enables these customizations to be easily applied to communities, collections or the entire repository.

This presentation demystifies the process of creating a Manakin theme and adapting it to the unique needs of your institutional repository. The presentation will be structured into four main sections: theme components, basic and complex theme development techniques, and an overview of advanced topics.

### **Theme components**

First, the presentation will cover the three components essential to a Manakin theme. The first and most essential component is a Cocoon sitemap that defines the theme, including any resources it will use. The XSL transformation is the component that creates a final output for the browser. The final component is a set of CSS and supporting files for styling that output.

### **Tiered theme development**

Theme development in Manakin is divided in tiers according to the technical skill-level necessary to modify a theme at that level; a *development* tier which requires Java and Cocoon experience, a *theme* tier which requires XML and XSL knowledge, and finally a *style* tier, which only requires an understanding of basic HTML and CSS.

### **Basic theme development (style tier)**

Next, basic development is discussed, including the low-level details necessary to create a theme: creating a new folder and sitemap, setting up a reference to the existing XSL templates library, copying one of the reference CSS files, and installing the theme in the `xmlui.xconf` file.

### **Complex theme development (theme tier)**

This section will go into greater detail about the XSL templates that create the final output for the browser. Examples of modifications possible at this stage include changes to the community or collection home page, or significantly altering the way an item is rendered.

### **Advanced topics**

Time permitting, the following special topics will be covered: extension of existing metadata handlers and creation of new ones, static page insertion, and utilization of non-XHTML output formats and non-XSL-based conversion mechanisms.

This presentation is the second in a set of three about the Manakin project. This presentation covers how to use themes to customize Manakin at your institution. Other presentations in the set introduce Manakin and its major architectural components and present a specific use case of using Manakin to express spatial metadata and complex objects.