Mirage XMLUI Theme and Restructured dri2xhtml for DSpace 1.7

Mirage XMLUI Theme and Restructured dri2xhtml for DSpace 1.7

@mire contribution of new theme to support easier development and branding. Provides the following:

1. Mirage, a new basic theme for the Manakin XML User Interface that contains the templates that need customization most often in the UI.
2. A Restructured dri2xhtml base templates for easing locating specific XSLT templates.

Mirage XMLUI Theme

Overall Benefits

- Clean new look and feel.
- Increased browser compatibility. The whole theme renders perfectly in today's modern browsers (Internet Explorer 7 and higher, Firefox, Safari, Chrome, ...)
- Easier to customize.

Enhanced Performance

Technical Features

Look & Feel

- The Simple Item Display underwent a full redesign to provide visitors with a clearer overview of available metadata and associated files.
- Item list views can now be displayed in two distinctive styles. Switching between these styles is possible with the new dspace.cfg parameter 'xmlui.theme.mirage.item-list.emphasis'.
  - The 'metadata' list style includes item abstracts in the listing and is suited for scientific articles.
  - The 'file' list style immediately shows you whether files are attached to the items, by displaying a large thumbnail icon for each of the items.

Structural enhancements for easier customization.

- Based on the new restructured dri2xhtml base templates. Templates in the theme, overriding the new base templates, are located in the same folder hierarchy to ensure full transparency.
- Automated browser feature detection for improved browser compatibility.
  - In other themes, user agent detection is used to identify which browser version your user is using. Based on the result of this detection, the theme would use a different cascaded style sheet (CSS) to render a compatible page for the visitor. This approach has 2 major issues:
    - User agent detection isn't very reliable
    - Maintaining these different CSS files is a maintenance nightmare for developers, especially when using features from newer browsers.
  - Mirage applies two novel techniques to resolve these issues.
    - For compatibility with older Internet Explorer browsers, conditional comments give the body tag a class corresponding to the version of IE.
    - Modernizr is used to detect which CSS features are available in the user's browser. This way you can target all browsers that support a certain feature using CSS classes, and rules affecting the same element can be put together in the same place for all browsers.
- CSS files are now split up according to function instead of browser. style.css will now fit most needs for customization. Following additional CSS files are included, but will rarely need to be changed:
  - reset.css ensures that browser-specific initializations are being reset.
  - base.css contains a few base styles
  - helper.css contains helper classes to deal with specific functionality.
  - handheld.css and print.css enable you to define styles for handheld devices and printing of pages.
- jQuery and jQueryUI are included by default. To avoid conflicts the authority control javascript has been rewritten to use jQuery instead of Prototype and Script.aculo.us.

Enhanced Performance

- Concatenation and Minification techniques for css and js files.
  - The IncludePageMeta has been extended to generate URL's to the concatenated version of all css files using the same media tag.
  - The ConcatenationReader has been created to return concatenated and minified versions of the css and js files.
  - Once js and css files have been minified and concatenated, they are being properly cached. As a result, the minification and concatenation operations only need to happen once, and do not include performance overhead.
  - Caution: when minification is enabled, all code-comments will be removed. This could be a problem for comments containing copyright notices, so for files with those comments you should disable minification by adding '?nominify' after the url e.g.
    
    `<map:parameter name="javascript" value="/lib/js/jquery-ui-1.8.5.custom.min.js?nominify"/>
    `Disabled by default, these features need to be enabled in the configuration using the properties 'xmlui.theme.enableConcatenation' and 'xmlui.theme.enableMinification'.
  - These features can be enabled for other themes as well, but will require an alteration of the theme's sitemap.
• Javascript references are included at the bottom of the page instead of the top. This optimizes page load times in general.

Screenshots
dri2xhtml-alt


Configuration and Installation

The alternative basic templates has been committed, and is currently dubbed "dri2xhtml-alt". Any of the existing themes can be updated to reference this new set of templates by replacing in your theme.xsl:

```xml
<xsl:stylesheet xmlns:i18n="http://apache.org/cocoon/i18n/2.1"
                xmlns:dri="http://di.tamu.edu/DRI/1.0/"
                xmlns:mets="http://www.loc.gov/METS/"
                xmlns:xlink="http://www.w3.org/TR/xlink/"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
                xmlns:dim="http://www.dspace.org/xmlns/dspace/dim"
                xmlns:xhtml="http://www.w3.org/1999/xhtml"
                xmlns:mods="http://www.loc.gov/mods/v3"
                xmlns:dc="http://purl.org/dc/elements/1.1/"
                xmlns="http://www.w3.org/1999/xhtml"
                exclude-result-prefixes="i18n dri mets xlink xsl dim xhtml mods dc">

<!-- comment out original dri2xhtml
<xsl:import href="../dri2xhtml.xsl"/>
and enable dri2xhtml-alt
-->

<xsl:import href="../dri2xhtml-alt/dri2xhtml.xsl"/>

<xsl:output indent="yes"/>
```

Because the contents of dri2xhtml-alt is identical to the current dri2xhtml.xsl and its derivatives, updating any of the existing themes to reference the new dri2xhtml-alt should not impose any changes in the rendering of the pages.
Features

- No changes to existing templates found in legacy dri2xhtml
- Drops inclusion of Handlers other than DIM and Default
- Templates divided out into files so they can be more easily located, divided by Aspect, Page and Functionality

/dspace-xmlui/dspace-xmlui-webapp/src/main/webapp/themes/dri2xhtml-alt/

aspect
  administrative
    harvesting.xsl
artifactbrowser
  COinS.xsl
  ORE.xsl
    artifactbrowser.xsl
  collection-list.xsl
  collection-view.xsl
common.xsl
  community-list.xsl
  community-view.xsl
item-list.xsl
item-view.xsl

general
  choice-authority-control.xsl

core
  attribute-handlers.xsl
  elements.xsl
  forms.xsl
  global-variables.xsl
  navigation.xsl
  page-structure.xsl
  utils.xsl
  dri2xhtml.xsl