

HowToFullyCustomMenus

Why?

Sometimes a site design calls for a completely different approach to a menu system that the can't do using the Lenya menu or fancy CSS. For example, I had a site that had three sections (The Products, The Services, The Legend) that needed to be displayed differently on the homepage only (different part of screen, displaying all immediate subpages but nothing below that) from the rest of the site sections.

So I wrote a system that looks for the three "special" sections and displays them differently. I though I would share it with the planet, hope it helps you.

Note this method does not destroy the original Lenya menu so you can still use it elsewhere in the site.

Handy resources

- <http://www.w3schools.com/xsl/> - Tutorial on XSL
- <http://www.w3schools.com/xpath/> - Tutorial on Xpath Queries

How?

1. Edit your publication-sitemap.xmap thusly, starting about line 124:

```
+ <map:part src="cocoon://navigation/{2}/{3}/breadcrumb/{5}.xml" />
  <map:part src="cocoon://navigation/{2}/{3}/tabs/{5}.xml" />
  <map:part src="cocoon://navigation/{2}/{3}/menu/{5}.xml" />
  <map:part src="cocoon://navigation/{2}/{3}/search/{5}.xml" />
  <map:part src="cocoon://navigation/{2}/{3}/home_menu/{5}.xml" />
  <map:part src="cocoon:/lenya-document-{1}/{3}/{4}/{page-envelope:document-path}" />
```

2. Make a new file at \$LENYA_HOME/build/lenya/webapp/lenya/pubs/[your pub]/lenya/xslt/navigation called home_menu.xsl
3. Write some XSL, hack around with the XSL from \$LENYA_HOME/build/lenya/webapp/lenya/xslt/navigation/menu.xsl, or hack around with my example (below) from the scenario (above).

```

<?xml version="1.0" encoding="UTF-8" ?>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:nav="http://apache.org/cocoon/lenya/navigation/1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:fn="http://www.w3.org/2005/02/xpath-functions"
  exclude-result-prefixes="nav"
  >
  <xsl:template match="nav:site">
    <div id="home_menu">
      <xsl:apply-templates select="nav:node" />
    </div>
  </xsl:template>
  <xsl:template match="nav:node[@href='products.html']">
    <div class="menu">
      <div id="menu1_header"><h2>The Products</h2></div>
      <br clear="left" />
      <ul>
        <xsl:apply-templates select="nav:node" />
      </ul>
    </div>
  </xsl:template>
  <xsl:template match="nav:node[@href='services.html']">
    <div class="menu">
      <div id="menu2_header"><h2>The Services</h2></div>
      <br clear="left" />
      <ul>
        <xsl:apply-templates select="nav:node" />
      </ul>
    </div>
  </xsl:template>
  <xsl:template match="nav:node[@href='legend.html']">
    <div class="menu">
      <div id="menu3_header"><h2>The Legend</h2></div>
      <br clear="left" />
      <ul>
        <xsl:apply-templates select="nav:node" />
      </ul>
    </div>
  </xsl:template>
  <xsl:template match="nav:node">
    <xsl:choose>
      <xsl:when test="parent::nav:node[@href = 'products.html']">
        <li><a href="{@href}"><xsl:apply-templates select="nav:label" /></a></li>
      </xsl:when>
      <xsl:when test="parent::nav:node[@href = 'services.html']">
        <li><a href="{@href}"><xsl:apply-templates select="nav:label" /></a></li>
      </xsl:when>
      <xsl:when test="parent::nav:node[@href = 'legend.html']">
        <li><a href="{@href}"><xsl:apply-templates select="nav:label" /></a></li>
      </xsl:when>
    </xsl:choose>
  </xsl:template>

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()" />
    </xsl:copy>
  </xsl:template>

</xsl:stylesheet>

```

4. In your \$LENYA_HOME/build/lenya/webapp/lenya/pubs/[your pub]/xslt/page2xhtml-homepage.xsl, add the line

```
<xsl:apply-templates select="xhtml:div[@id = 'home_menu']" />
```

where you want the new menu to be.

"In This Section" Menu

Here is a useful menu I wrote. It lists all the pages immediately below the current page in the heirarchy under the heading "In This Section". If there are no pages immediately below the current page, it disappears. This is an extremely useful bit of code. Install it using the method above.

```

<!DOCTYPE xsl:stylesheet [
<!ENTITY nbsp '&#160; '
]>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:nav="http://apache.org/cocoon/lenya/navigation/1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  exclude-result-prefixes="nav"
>

<xsl:template match="nav:site">
  <xsl:apply-templates select="nav:node"/>
  &nbsp;
</xsl:template>

<xsl:template match="nav:node[@current='true']">

  <xsl:if test="count(nav:node) > 0">
    <div id="sub_tabs">
      <div class="tab_header">In this section: </div><br/>
      <xsl:for-each select="nav:node">
        <xsl:choose>
          <xsl:when test="@visibleinnav = 'false'"/>
          <xsl:otherwise>
            <xsl:if test="position() > 1">
              <xsl:call-template name="separator"/>
            </xsl:if>
            <xsl:call-template name="tab"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:for-each>
    </div>
  </xsl:if>

</xsl:template>

<xsl:template match="nav:node">
  <xsl:apply-templates select="nav:node"/>
</xsl:template>

<xsl:template name="tab">
  <div class="tab"><a href="{@href}"><xsl:value-of select="nav:label"/></a></div>
</xsl:template>

<xsl:template name="separator">
  <!--
  <div class="separator"> ... </div>
  -->
</xsl:template>

<xsl:template match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>

```