

ProposalXIncludeAggregation

Proposal: Enabling XInclude to implement page fragments

Note: This is just an idea. It has been briefly discussed on the lenya-dev mailing list and was found to be one of several possible ways to achieve the goal. It is also work in progress in that it is not yet compatible with the authoring / live mechanism in Lenya.

Most modern websites have a complex layout which in many cases is not just navigation and content, but the content is usually divided into main content and additional elements. (FixMe: Find a good example!)

(Take this further ...)

Adding the XInclude Transformer to the lenybody-pipeline

Add the XInclude Transformer into the appropriate pipeline in `publication-sitemap.xmap`:

```
<!-- This is the pipeline that builds the page. It aggregates all
the navigational elements (breadcrumb, tabs, menu) with the actual
content of the document. -->
<map:pipeline>
  <!-- /lenyabody-{rendertype}/{publication-id}/{area}/{doctype}/{url} -->
  <map:match pattern="lenyabody-*/*/*/*/**">
    <map:aggregate element="cmsbody">
      <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:/lenya-document-{1}/{3}/{4}/{page-envelope:document-path}"/>
    </map:aggregate>

    <!-- start inserting here -->
    <map:transform type="xinclude" />
    <!-- end inserting here -->

    <map:transform src="xslt/page2xhtml.xsl">
      <map:parameter name="root" value="{page-envelope:context-prefix}/{2}/{3}"/>
      <map:parameter name="url" value="{5}"/>
    </map:transform>
    <map:serialize type="xml"/>
  </map:match>
</map:pipeline>
```

Note: It might be an alternative to have a custom "withincludes" doctype and to move the XInclude transformer into the local pipeline for only that doctype. This might save passing the XInclude transformer for pages in which there is nothing included at all. It might also be worth looking at the CInclude transformer instead.

Using xi:include in a document

Make sure you have defined the xi namespace in your document:

```
xmlns:xi="http://www.w3.org/2001/XInclude"
```

`xslt/mount-href.xslt`

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns="http://www.w3.org/1999/xhtml"
>

<xsl:param name="mount" />

<xsl:template match="xi:include">
  <xi:include>
    <xsl:attribute name="href"><xsl:value-of select="$mount"/><xsl:value-of select="@href"/></xsl:attribute>
  </xi:include>
</xsl:template>

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

</xsl:stylesheet>
```

See also:

- [GetPageWalkthrough](#)
- [PublicationSitemapEntryPoints](#)