ALLAN THRAEN | 16 years ago | PDF |

XSLT IN EPISERVER CMS 5

Personally I'm not a big fan of neither XSLT nor XML. In fact, my feelings around XML is expressed in this quote I heard recently: "XML is like children. They start out cute and small, then they grow..." (I don't remember who said it - if it was YOU, mail me and tell me to credit you for those words of wisdom). In my opinion XSLT's are mainly just good for job-security for XSLT developers - they are about as friendly to read as regular expressions - and terrible to maintain. Nevertheless a lot of people like them due to the way they help separate design from data - and I've already been asked the question "how can I work with XSLT in EPiServer" many times. So, now I thought I better do something about it, so one dark and cold evening I made a web control that hopefully will satisfy all the XSLT magicians out there!

The control will create a XML document representing the current page that has this structure:

```
<page>
    <properties>
        <property name="PageLink" type="PageReference" isdynamic="False" isnull="False">3</property>
        ...
    </properties>
    <children>
        <page>
            ...
        </page>
    </children>
</page>
```

and then transform that XML using the XSLT you provide.

In order to use the control, you'll need to place the dll in the "bin" folder, and register it on the page you wish to use it on. Then you can put it on the page like this:

```
<research:XSLT runat="server" id="xslt2" MaxChildDepth=1 IncludeDynamic="true">
 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns="http://www.
 <xsl:output method="html"/>
 <xsl:template match="page">
     <h1><xsl:value-of select="properties/property[@name='PageName']"/></h1>
     <ul>
     <xsl:for-each select="properties/property">
         <li><xsl:value-of select="@name"/>: <xsl:value-of select="."/></li>
     </xsl:for-each>
     <li>
     Children:
         <xsl:apply-templates select="children/page"/>
     </li>
     </ul>
     <br />
 </xsl:template>
 </xsl:stylesheet>
</research:xslt>
```

In the above case the XSLT is specified within the controls tag, but you can also reference an external xslt file, by setting the property "TransformationFile" to the url of the XSLT file. Or - if you're feeling mean - you can bind the XSLT contents to a LongString property in EPiServer and let the editors figure out the extended stylesheets - that'll freak them out for sure :-)

By default the control will begin with rendering the XSLT on the current page data - if you want to base it on another page, it can be set up in well-known "PageLink" and "PageLinkProperty" properties. The "IncludeDynamic" property specifies whether to include dynamic properties or not, and the "AutoHtmlDecode" property specifies if html-tags within properties should be rendered as HTML tags or as text on the page.

Enjoy!

P.S. If you insist on playing around with XSLT and use this control, I wrote a XPathChecker some time ago that might come in handy.

The license for the control is as follows:

EPiOpen License terms 1.0

This software is unsupported and provided "as is".
You can use this software free of charge AND you can modify the software and sell it as your own software AS LONG AS THE SOFTWARE RUNS ON A COMPUTER WITH A VALID EPISERVER LICENSE. Any modified software cannot be named similar/same as the original".

EPiServer AB is not to be held responsible in any way for any damage or loss of income and/or data due to the use of the software.

RECENT POSTS

Phone: +45 26 13 66 96
CVR: 39680688

in ☺

Phone: +45 26 13 66 96
CVR: 39680688

in ☺