Subject: Re: xsl transformation slow
Posted by braingeyser on Mon, 19 Mar 2007 15:18:15 GMT
View Forum Message <> Reply to Message

Just in case my ramblings have confused you a little, here is my final version of "display_horizontal"

```
<!-- display details horizontally -->
<xsl:template name="display_horizontal">
  <xsl:param name="zone"/>      <!-- could be 'main', 'inner', 'outer', etc -->
  <xsl:param name="table_record" />
  <xsl:param name="multiple"/>  <!-- set this for more than one occurrence -->

  <xsl:variable name="table" select="name()"/>          <!-- current table name -->
  <xsl:variable name="position" select="position()"/>   <!-- current row within table -->

  <tr>
    <!-- set the row class to 'odd' or 'even' to determine the colour -->
    <xsl:attribute name="class">
      <xsl:choose>
        <xsl:when test="position()mod 2">odd</xsl:when>
        <xsl:otherwise>even</xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>

 <!-- step through the fields defined in the STRUCTURE element -->
    <xsl:for-each select="/root/structure/*[name()=$zone]/row/cell[@field]">

      <!-- get fieldname from the FIELD attribute -->
      <xsl:variable name="fieldname" select="@field" />

      <!-- select the field (identified in STRUCTURE) from the current row of the specified table -->
      <xsl:variable name="field" select="$table_record/*[name()=$fieldname]" />

      <td>
        <!-- process the named field from the current row -->
        <xsl:call-template name="datafield">
          <xsl:with-param name="item"     select="$field"/>
          <xsl:with-param name="itemname" select="$fieldname"/>
          <xsl:with-param name="path"     select="$table"/>
          <xsl:with-param name="multiple" select="$multiple"/>
          <xsl:with-param name="position" select="$position"/>
        </xsl:call-template>
      </td>

    </xsl:for-each>
  </tr>
```

</xsl:template> <!-- display_horizontal -->

and this is the change needed in the calling scripts to make it work...

std.link1.xsl

```
            <!-- process each non-empty row in the MAIN table of the XML file -->
            <xsl:for-each select="//*[name()=$main][count(*)&gt;0]">

              <!-- display all the fields in the current row -->
              <xsl:call-template name="display_horizontal">
                 <xsl:with-param name="zone" select="'main'"/>
     <xsl:with-param name="table_record" select="." /> <!-- ADD THIS LINE TO ALL CALLING
SCRIPTS -->
              </xsl:call-template>
            </xsl:for_each>
```

similar change in std.link1.xsl

```
          <!-- process each non-empty row in the INNER/CHILD table of the XML file -->
            <xsl:for-each select="//*[name()=$link][count(*)&gt;0]">

              <!-- display all the fields in the current row -->
              <xsl:call-template name="display_horizontal">
                <xsl:with-param name="zone"     select="'link'"/>
   <xsl:with-param name="table_record" select="." />
                <xsl:with-param name="multiple" select="'y'"/>
              </xsl:call-template>

            </xsl:for-each>
```

etc etc...

Seems to work correctly. And it's lightning fast. Give it a whirl, what's the worst that could happen?