Emetsger__StyleSheets__StyleTableRows

<?xml version="1.0" encoding="utf-8"?>
<html>
You could probably package up the XSL in a jar and use a resource:// to resolve the stylesheet. You could probably also run this as its own aspect.

theme sitemap

```
<map:transform src="tableclass.xsl"/>
```

<theme directory>/tableclass.xsl

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

<xsl:output indent="yes"/>

<!--

Replace the 'class' attribute on <tr> elements that don't have 'even' or 'odd' in their values.
This is so rows can be styled properly when there are a variable number of rows in a table.
Like when displaying optional metadata fields.
-->

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

<!--
Override the copy for <tr> elements that have a 'class' attribute with a value that does not
contain the text 'odd' or 'even'
-->
<xsl:template match="//xhtml:tratclass and not(contains(atclass, 'odd')) and not(contains(atclass, 'even'))">

<xsl:variable name="parity">
<xsl:choose>
<!-- <table>s with a 'ds-includeSet-table' class: the first row's
class is 'even' and the second is 'odd' and so on. So we
accomodate this... -->
<xsl:when test="position() mod 2 = 0">odd</xsl:when>
<xsl:otherwise>even</xsl:otherwise>
</xsl:choose>
</xsl:variable>

<xsl:variable name="newclass" select="concat(@class, ' ', $parity)"/>

<!-- Output the row, replacing the original value of the 'class' attribute
with the new value containing the parity. Simply copy everything else. -->
<xsl:element name="tr">
<xsl:for-each select="@*">
<xsl:choose>
<xsl:when test="name(.) = 'class'">
<xsl:attribute name="class"><xsl:value-of select="$newclass"/></xsl:attribute>
</xsl:when>
<xsl:otherwise>
<xsl:copy-of select="."/>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:copy-of select="node()"/>
</xsl:element>
</xsl:template>

</xsl:stylesheet>
```

</html>