

## Tips on modifying the ATOC-1 stylesheet to suit your needs

The general design principle behind the stylesheet is to copy everything from the source document to the target document and to add or modify specific elements and attributes as required to suit the needs of the repository.

Most EAD elements (and their corresponding attributes) can be used at multiple levels in the hierarchical structure of the document. So if you want to target the changes to a particular level of the hierarchy, you'll need to use a [] predicate to qualify the expression or you can use the "mode" attribute within <xsl:template> to narrow the choice of elements selected. Both of these techniques are used in the stylesheet.

Important note: In its original form, the stylesheet targets specific changes to elements and attributes within the high-level <did> **(mode="hl-did")** and the <dsc> section **(mode="dsc")**. These sections are delineated in the stylesheet. If you want to target additional changes to elements and attributes within the scope of these sections, you must use the corresponding "mode" attribute on any new template rules that you devise.

For instance, if you want add an `ENCODINGANALOG` attribute to the <unitid> element in the high-level <did>, you would devise your template rule like this:

```
<xsl:template match="unitid" mode="hl-did">
<xsl:copy>
<xsl:copy-of select="@*"/>
<xsl:attribute name="encodinganalog">099</xsl:attribute>
<xsl:apply-templates mode="hl-did"/>
</xsl:copy>
</xsl:template>
```

Note that the mode attribute is used on the <xsl:template> and <xsl:apply-templates> elements.

As stated, the mode attribute is only required when targeting changes to elements and attributes within the high-level <did> and the <dsc> sections. Outside of these two specific cases (or if you create new ones),

you can target changes to a single element by using a predicate to narrow the scope to that element.

For instance, if you want to target changes to the <scopecontent> element that is a direct child of <archdesc>, you could use a rule like this:

```
<xsl:template match="scopecontent[parent::archdesc]">
```

This says, match the <scopecontent> element whose immediate parent element is <archdesc>. There is only one place in your EAD document where this test can be true, so only that one instance of <scopecontent> is matched, leaving all other <scopecontent> elements untouched.

On the other hand, if you want to target changes to all elements of a specific name (other than those that occur in the high-level <did> and <dsc>), then just leave the predicate off:

```
<xsl:template match="phystech">
<xsl:copy>
<xsl:copy-of select="@*" />
<xsl:attribute name="encodinganalog">340</xsl:attribute>
<xsl:apply-templates />
</xsl:copy>
</xsl:template>
```

## Adding attribute values

The <xsl:copy> element copies the element matched in the template rule and the <xsl:copy-of select="@\*" /> copies all of its attributes. If you want to add an attribute, you can do that after the <xsl:copy> or <xsl:copy-of select="@\*" /> element:

```
<xsl:copy>
<xsl:copy-of select="@*" />
<xsl:attribute name="encodinganalog">340</xsl:attribute>
<xsl:attribute name="id">a35</xsl:attribute>
```

There are numerous examples of this within the stylesheet. Just follow the pattern.

Feel free to contact me if you are having problems or need help modifying the stylesheet.

carlsonm@u.washington.edu