OmniUpdate Training Tuesday
Extending Snippets and Components with XSL

Zoom Event # 601 783 255

Audio will be heard on your computer speakers.

If you do not have working computer speakers, call 1-408-792-6300. Enter event number and attendee ID or press # if no attendee ID.

Webcast will start at the top of the hour.

Presented By:
Robert Kiffe
Sr. Technical Support Engineer
▪ What’s the Challenge?
▪ Introducing Components
▪ Table Transformations
▪ XSL Concepts Review
▪ 3-Step Strategy
▪ Tips and Tricks
▪ Hands-On
▪ Q&A
What’s the Challenge?

- Some design elements are too difficult to maintain directly in the WYSIWYG

Example (from getbootstrap.com):

**Progress Bar with Label**

Add labels to your progress bars by placing text within the `.progress-bar`.

```html
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100">25%</div>
</div>
```
‘Components’ provide a form-based method for entering content

Produce complex HTML output without any custom scripts
‘Tables’ provide a form-like method for storing content

Tables are a robust element in the WYSIWYG and content reorganization is fairly easy

Requires XSL to ‘transform’ content into desired HTML output
The Power of XSL

- XSL ~ server-side language
- Traditionally used to convert XML data into another XML data structure
- Now used to ‘transform’ XML to HTML
XSL: Apply-Templates

- Core functionality of XSL
- Initiated by an “Apply-Templates” instruction
- Iterated content is ‘captured’ by “Template Rules” that match the current iteration

XML
<pre>
<profiles>
  <profile>
    <name>Bruce Wayne</name>
    <email>notbatman@wayneindustries.com</email>
  </profile>
  <profile>
    <name>Tony Stark</name>
    <email>iamironman@starkindustries.com</email>
  </profile>
</profiles>
</pre>

XSL
<pre>
<xsl:stylesheet version="1.0">
  <xsl:template match="profile">
    <li>
      <xsl:value-of select="/name" />
    </li>
  </xsl:template>
</xsl:stylesheet>
</pre>
More than one ‘Template Rule’ can match any given item

Follows basic ruleset:

1. ‘Mode’ attribute must match
2. XSL import precedence
3. ‘Priority’ attribute value (Largest value goes first)
4. ‘Match’ expression specificity (similar to CSS)

```xml
<profiles>
  <profile>
    <name>Bruce Wayne</name>
    <email>notbatman@wayneindustries.com</email>
  </profile>
  <profile>
    <name>Tony Stark</name>
    <email>iamironman@starkindustries.com</email>
  </profile>
</profiles>
```

```xsl
<xsl:template match="profile">
  <li>
    <xsl:value-of select="./name" />
  </li>
</xsl:template>

<xsl:template match="profile[name='Bruce Wayne']">
  <li>
    ^O^ - <xsl:value-of select="./name" />
  </li>
</xsl:template>
```
1. Data Structure
   • Component Form Builder
   • Snippet (usually a Table)

2. Example Output Code
   • Component Source Code
   • (And/Or) Template Match

3. Update #2 with data/content from #1
Tips and Tricks

▪ Components
  ▪ Output as XML for easy XSL processing
  ▪ Output as HTML, but use XSL to process ‘child’ content

▪ Table Transformations
  ▪ Choose a table layout that best matches the content
  ▪ Add custom styling to your WYSIWYG CSS to make the table appear more like a form
  ▪ See our sample code: https://github.com/omniupdate/example-code-table-transformations

```xml
<xsl:template match="div[@class='card']//img[@href='']" />
```

```xml
<ouc:div label="maincontent" group="Everyone" button-text="Main Content">
  <ouc:editor csspath="/workshop-components-xsl/editor/wysiwyg.css"
             cssmenu="/workshop-components-xsl/editor/styles.txt"/>
```

Let’s Take a Look!
NEXT TRAINING TUESDAY

Next month’s Training Tuesday will be held on MAY 28TH

We’ll be covering the topic of CONFIGURATION FILES.

Be sure to visit the OCN or Support Site for further details!

http://ocn.omniupdate.com
http://support.omniupdate.com
Thank you!