



OmniUpdate Training Tuesday

XSL 3.0



Presented By:
Cory Madden

Web Developer – Team
Lead

Zoom Event # **220-908-504**

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.



AGENDA

- expand-text
- Function Chaining
- Concatenation Shorthand
- xsl:mode
- Demo



expand-text

- May appear on any element in the XSL
- Only valid in XSL 3.0
- Value can be “yes” or no”
- Determines whether nodes below it will be similar to xsl:value-of
- Be careful when trying to output CSS / JavaScript when expand-text is set to “yes”



expand-text Examples

- Old way

```
<p><xsl:value-of select="$heading" /></p>
```

```
<ul>  
  <xsl:for-each select="1 to 10">  
    <li><xsl:value-of select="position()" /></li>  
  </xsl:for-each>  
</ul>
```

- New Way

```
<p xsl:expand-text="yes">{$heading}</p>
```

```
<ul>  
  <xsl:for-each select="1 to 10" expand-text="yes">  
    <li>{position()}</li>  
  </xsl:for-each>  
</ul>
```



Function Chaining

- Known as arrow operator
- Only valid in XSL 3.0
- Using this will apply the function to the value of an expression, using the value as the first argument to the function
- Helps you stay organized with parentheses



Function Chaining Example

```
<xsl:variable name="example-1">Value1, Value2, Value3</xsl:variable>
```

- Old Way

```
<ul>  
  <xsl:for-each select="tokenize(lower-case($example-1), ', ')">  
    <li><xsl:value-of select="normalize-space(.)" /></li>  
  </xsl:for-each>  
</ul>
```

- New Way

```
<ul>  
  <xsl:for-each select="$example-1 => lower-case() => tokenize(', ')">  
    <li><xsl:value-of select="normalize-space(.)" /></li>  
  </xsl:for-each>  
</ul>
```



Concatenation Shorthand

- A way to "shorten" your concatenation statements
- Only valid in XSL 3.0
- Good to use for short concatenation but sometimes is better to use the concat()
- Does cause confusion because it is " || " which is "or" in other programming languages



Concatenation Shorthand Examples

```
<xsl:variable name="var-one">Cory</xsl:variable>  
<xsl:variable name="var-two">Madden</xsl:variable>  
<xsl:variable name="var-three">OmniUpdate</xsl:variable>
```

- Old Way

```
<p><xsl:value-of select="concat($var-one, ' ', $var-two, ' ', $var-three)" /></p>
```

- New Way

```
<p><xsl:value-of select="$var-one || ' ' || $var-two || ' ' || $var-three" /></p>
```




- A mode is way to separate xsl:template-matches during xsl:apply-templates
- Only Valid in XSL 3.0
- Allows for easy way to define different modes
- Different Types of copy effects
- We primarily use “shallow-copy” option
- Old way for defining modes was a bit complicated
- New way is simply one line!



xsl:mode Example

- Old Way

```
<xsl:template match="attribute()|text()|comment()" mode="first-mode">  
  <xsl:copy />  
</xsl:template>
```

```
<xsl:template match="element()" mode="first-mode">  
  <xsl:element name="{name()}">  
    <xsl:apply-templates select="attribute()|node()" mode="#current"/>  
  </xsl:element>  
</xsl:template>
```

- New Way

```
<xsl:mode on-no-match="shallow-copy" name="first-mode" />
```



Let's Take a Look!



NEXT TRAINING TUESDAY

Next month's Training Tuesday will be held on **APRIL 30TH**.

We'll be covering the topic of **ADVANCED COMPONENTS**.

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

<http://ocn.omniupdate.com>

<http://support.omniupdate.com>



Thank you!