



This document describes changes to WebVoyage and other configuration files from Voyager 10.0.0 to Voyager 10.1.0. Documented below are the files changed and the specific changes to each file.

Please see the Voyager 10.1.0 defect fixes for specific bug fixes addressed by these files. This document can be found on the Knowledge Center under Home > Voyager > Release Notes > 10.1.0

WebVoyage changes

The upgrade to 10.1.0 will overwrite `exl_default` directory with all of the new files listed below, as well as the default versions of all other WebVoyage skin files in that directory. **Never make changes to files in `exl_default`. Make copies of the files to update other skins.** Review skin file changes and do one of the following:

- If a changed file has not been customized in your custom skin, replace it with a new copy from `exl_default`.
- If a file has been customized in your custom skin, review changes below, make a copy of the file from `exl_default`, and *recommit customizations to the copy of the new file*.

Test the skin after each customization is reapplied to the new files. See [“What is process for upgrading skin directories?”](#) for additional details.

Files with a `.tml` extension do not need to be customized – these are templates and are not actively used in the OPAC display.

If upgrading from a version of Voyager prior to 10.0.0, please note that there is not a Changed Skin Files document. Since the process for upgrading to Voyager 10.0.0 and higher from an earlier release requires that sites begin with the Voyager 10.x version of all skin files, no list of changed files is necessary.

Changed skin files from 10.0.0 to 10.1.0

Changed Files:

```
tomcat/vwebv/context/vwebv/ui/exl_default/jscripts/googleBooksAvail.js
tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/configs/export/endNoteCitationConfig.xml
tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/display/display.xsl
tomcat/vwebv/context/vwebv/ui/exl_default/xsl/userTextConfigs/limits.xml
tomcat/vwebv/context/vwebv/ui/exl_default/xsl/userTextConfigs/pageProperties.xml
```

Configuration file changes

No files changed for Voyager 10.1.0

File Content Changes 10.0.0 to 10.1.0:

Below, “Original lines” refers to the file contents in 10.0.0 “New lines” refers to the new content in 10.1.0 Line numbers in customized skins may vary based on local customizations.

Changed File tomcat/vwebv/context/vwebv/ui/exl_default/jscripts/googleBooksAvail.js

Original lines 147 - 154

```
// and the callback funtion is also specified as a URI argument.
var scriptElement = document.createElement("script");
scriptElement.setAttribute("id", "googleBooksScript");
scriptElement.setAttribute("src", "http://books.google.com/books?bibkeys=" +
escape(document.getElementById(bibKey).value) + "&jscmd=viewapi&callback="+handler);
scriptElement.setAttribute("type", "text/javascript");
```

```
// make the request to Google booksearchsh
```

New lines 147 - 154

```
// and the callback funtion is also specified as a URI argument.
var scriptElement = document.createElement("script");
scriptElement.setAttribute("id", "googleBooksScript");
scriptElement.setAttribute("src", "https://books.g!oogle.com/books?bibkeys=" +
escape(document.getElementById(bibKey).value) + "&jscmd=viewapi&callback="+handler);
scriptElement.setAttribute("type", "text/javascript");
```

```
// make the request to Google booksearchsh
```

Changed File tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/configs/export/endNoteCitationConfig.xml

Original lines 28 - 40

```
<displayTag field="710" indicator1="X" indicator2="X" subfield="ab"/>
</displayTags>

<titleTags label="Title:">
<displayTag field="130" indicator1="X" indicator2="X" subfield="aplskfmnor"/>
<displayTag field="240" indicator1="X" indicator2="X" subfield="aplskfmnor"/>
<displayTag field="730" indicator1="X" indicator2="X" subfield="aplskfmnor"/>
<displayTag field="245" indicator1="X" indicator2="X" subfield="abcfknps"/>
</titleTags>

<displayTags label="Series Editor:" >
<disp! layTag field="800" indicator1="X" indicator2="X" subfield="a" />
```

New lines 28 - 40

```
<displayTag field="710" indicator1="X" indicator2="X" subfield="ab"/>
</displayTags>

<displayTags label="Title:">
<displayTag field="130" indicator1="X" indicator2="X" subfield="aplskfmnor"/>
<displayTag field="240" indicator1="X" indicator2="X" subfield="aplskfmnor" preText="[" postText="]"/>
<displayTag field="730" indicator1="X" indicator2="X" subfield="aplskfmnor"/>
<displayTag field="245" indicator1="X" indicator2="X" subfield="ab! cfknps"/>
</displayTags>

<displayTags label="Series Editor:" >
<displayTag field="800" indicator1="X" indicator2="X" subfield="a" />
```

Changed File tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/display/display.xsl

Original lines 85 - 92

```
<xsl:variable name="bibFormats" select="$Configs/pageConfigs/bibFormats"/>
```

New lines 85 - 93

```
<xsl:variable name="bibFormats" select="$Configs/pageConfigs/bibFormats"/>
```

```
<xsl:variable name="browse" select="%2B"/>
```

```
<!-- ## Definition of URN and DOI link from webvoyage.properties file the default is a blank string ## -->  
<xsl:variable name="doiLink">
```

```
<!-- use the '+' instead of '%2B' to avoid double URL encoding -->
```

```
<xsl:variable name="browse" select="+"/>
```

```
<!-- ## Definition of URN and DOI link from webvoyage.properties file the default is a blank string ## -->  
<xsl:variable name="doiLink">
```

Changed File tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/display/display_text.xsl

Original lines 67 - 78

```
<xsl:choose>  
<xsl:when test="$recordType='bib'">  
<!-- ## Bibliographic Title ## -->  
<xsl:call-template name="buildTitle">  
<xsl:with-param name="mfhdID" select="$mfhdID"/>  
<xsl:with-param name="recordType" select="$recordType"/>  
</xsl:call-template>
```

```
<!-- ## Bibliographic Tags ## -->  
<xsl:for-each select="/display/displayTags">
```

Original lines 200 - 257

```
<!-- ##### -->
```

```
<xsl:template name="outputDisplayData">  
<xsl:param name="displayData"/>
```

```
<xsl:choose>  
<xsl:when test="string-length($displayData) and displayTag/@field='9000'">  
<xsl:choose>  
<xsl:when test="string-length(@label)">  
<xsl:value-of select="@label"/>  
</xsl:when>  
</xsl:choose>
```

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

```
&! It;xsl:call-template name="addTabsBeforeDataAfterLabel">  
<xsl:with-param name="label" select="@label"/>  
</xsl:call-template>
```

```
<xsl:copy-of select="$displayData"/>  
</xsl:when>
```

```
<xsl:when test="string-length($displayData)">  
<xsl:choose>  
<xsl:when test="string-length(@label)">  
<xsl:value-of select="@label"/>  
</xsl:when>
```

New lines 67 - 73

```
<xsl:choose>  
<xsl:when test="$record! dType='bib'">  
<!-- ## Bibliographic Tags ## -->  
<xsl:for-each select="/display/displayTags">
```

New lines 195 - 595

```
<!-- ##### -->
```

```
<!-- template called by outputDisplayData -->
```

```
<!-- from stack overflow use to get the last space in the data to break the line further -->  
<xsl:template name="last-index-of">  
<xsl:param name="txt"/>  
<xsl:param name="remainder! r" />  
<xsl:param name="delimiter" />
```

```
<xsl:choose>  
<xsl:when test="contains($remainder, $delimiter)">  
<xsl:call-template name="last-index-of">  
<xsl:with-param name="txt" select="$txt"/>  
<xsl:with-param name="remainder" select="substring-after($remainder, $delimiter)"/>  
<xsl:with-param name="delimiter" select="$delimiter"/>  
</xsl:call-template>  
</xsl:when>  
<xsl:otherwise>  
<xsl:variable name="lastIndex" select="string-length(substring($txt, 1, string-length($txt)-string-length($remainder)))+1"/>  
<xsl:choose>  
<xsl:when test="string-length($remainder)=0">  
<xsl:value-of select="string-length($txt)"/>  
</xsl:when>  
<xsl:when test="$lastIndex>0">  
<xsl:value-of select="($lastIndex - string-length($delimiter))"/>  
</xsl:when>  
<xsl:otherwise>
```

```

</xsl:choose>
<xsl:call-template name="addTabsBeforeDataAfterLabel">
<xsl:with-param name="label" select="@la! bel"/>
</xsl:call-template>
<xsl:copy-of select="$displayData"/>
<xsl:value-of select="$new_line"/>
</xsl:when>

<xsl:when test="not(string-length($displayData))">
<xsl:if test="string-length(@notFound)">
<xsl:choose>
<xsl:when test="string-length(@label)">
<xsl:value-of select="@label"/>
</xsl:when>
</xsl:choose>

<xsl:call-template name="addTabsBeforeDataAfterLabel">
<xsl:with-param name="label" select="@label"/>
<! ;xsl:call-template>

<xsl:value-of select="@notFound"/>
<xsl:value-of select="$new_line"/>
</xsl:if>
</xsl:when>
</xsl:choose>

```

```

<xsl:value-of select="0"/>
</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- from stack overflow recursively pad the space up to the max of each column as needed -->
<xsl:template name="fillSpace">
<xsl:param name="cnt"/>
! <xsl:param name="chr"/>
<xsl:value-of select="$chr"/>
<xsl:if test="$cnt &gt; 1">
<xsl:call-template name="fillSpace">
<xsl:with-param name="cnt" select="$cnt - 1"/>
<xsl:with-param name="chr" select="$chr"/>
</xsl:call-template>
</xsl:if>
</xsl:template>

<xsl:template name="removeLeadingTabsSpaces!">
! <xsl:param name="txt"/>
<xsl:param name="data"/>

<xsl:choose>
<xsl:when test="starts-with($data, $tab)">
<xsl:call-template name="removeLeadingTabsSpaces">
<xsl:with-param name="txt" select="$txt"/>
<xsl:with-param name="data" select="substring-after($data, $tab)"/>
</xsl:call-template>
</xsl:when>
<xsl:when test="starts-with($data, ' ')">
<xsl:call-template name="removeLeadingTabsSpaces">
<xsl:with-param name="txt" select="$txt"/>
<xsl:with-param name="data" select="substring-after($data, ' ')">
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:variable name="lastIndex" select="string-length(substring($txt, 1, string-length($txt)-string-length($data)))+1"/>
<xsl:choose>
<xsl:when test="string-length($data)=0">
<xsl:value-of select="string-length($txt)"/>
</xsl:when>
<xsl:when test="$lastIndex &gt; 0">
<xsl:value-of select="$lastIndex"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="0"/>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

```

```

<xsl:template name="outDisplayedString">
<xsl:param name="fullString"/>
<xsl:param name="curDisplayString"/>
<xsl:param name="columnWidth"/>
<xsl:param name="labelColumnWidth"/>
<xsl:!-- param name="curIndx"/>
<xsl:param name="padLabel"/>

<xsl:variable name="lastSpaceIndx" >
<xsl:choose>
<!-- this is the last section of the full string -->
<xsl:when test="(string-length($fullString) - ($curIndx + string-length($curDisplayString))) &lt;= 0">
<xsl:value-of select="string-length($curDisplayString)"/>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="last-inde! x-of">
<xsl:with-param name="txt" select="$curDisplayString" />
<xsl:with-param name="remainder" select="$curDisplayString" />
<xsl:with-param name="delimiter" select="" />
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<!-- indentation for second line of data as in classic -->
<xsl:choose>
<xsl:when test="$padLabel = 'yes&! #39;'">
<xsl:call-template name="fillSpace" >
<xsl:with-param name="cnt" select="$labelColumnWidth"/>
<xsl:with-param name="chr" select="" />
</xsl:call-template>
<xsl:value-of select="" />
</xsl:when>
</xsl:choose>
<xsl:choose>
<xsl:when test="($lastSpaceIndx = string-length($curDisplayString)) or ($lastSpaceIndx = 0)"><xsl:value-of
select="$curDisplayString"/>
</xsl:when>
<xsl:when test="$lastSpaceIndx &gt; 0">
<xsl:variable name="outStr">
<xsl:value-of select="substring($curDisplayString, 1, $lastSpaceIndx)"/>
</xsl:variable>
<xsl:value-of select="$outStr"/>
<xsl:choose>
<xsl:when test="$columnWidth &gt; string-length($outStr)">
<xsl:call-template name="! fillSpace" >
<xsl:with-param name="cnt" select="$columnWidth - string-length($outStr)"/>
<xsl:with-param name="chr" select="" />
</xsl:call-template>
</xsl:when>
</xsl:choose>
</xsl:when>

```

```

</xsl:choose>
<xsl:variable name="remaining">
<xsl:choose>
<xsl:when test="$lastSpaceIndx &gt; 0">
! <xsl:value-of select="substring($fullString, $curIndx + $lastSpaceIndx, string-length($fullString) - ($curIndx +
$lastSpaceIndx) + 1)"/>
</xsl:when>
</xsl:choose>
</xsl:variable>
<xsl:if test="string-length($remaining) &gt; 0">
<xsl:value-of select="$new_line"/>
<xsl:variable name="processedStr">
<xsl:choose>
<xsl:when test="string-length($remaining) &lt;= $columnWidth">
<xsl:value-of select="$remaining"/>
! </xsl:when>
<xsl:otherwise>
<xsl:value-of select="substring($remaining, 1, $columnWidth)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<!-- column width in second line will be indented -->
<xsl:call-template name="outDisplayedString">
<xsl:with-param name="fullString" select="$fullString"/>
<xsl:with-param name="curDisp! layString" select="$processedStr"/>
<xsl:with-param name="columnWidth" select="56"/>
<xsl:with-param name="labelColumnWidth" select="$labelColumnWidth"/>
<xsl:with-param name="padLabel" select="yes"/>
<xsl:with-param name="curIndx" select="($curIndx + $lastSpaceIndx)"/>
</xsl:call-template>
</xsl:if>
</xsl:template>

```

```

<xsl:template name="getOut! putLabel">
<xsl:choose>
<xsl:when test="string-length(@label)">
<xsl:value-of select="@label"/>
</xsl:when>
</xsl:choose>
</xsl:template>

```

```

<!-- as in the classic webvoyage, print out multiple lines of data -->
<!-- if it is longer than a line, pad data as necessary -->
<xsl:template name="outputData">
<xsl:param name="data"/>
<xsl:param name="hasLabel"/>
<xsl:param name="labelColumn! nWidth"/>
<xsl:param name="dataColumnWidth"/>

```

```

<!-- output the data and check if there is a second line, then -->
<!-- call the same template recursively with updated paramater -->

```

```

<!-- fill space if there is no label on the same line -->
<xsl:choose>
<xsl:when test="$hasLabel = 'no'">
<xsl:call-template name="fillSpace" >
<xsl:with-param name="cnt" select="$labelColumnWidth"! ;/>
<xsl:with-param name="chr" select="" />
</xsl:call-template>
</xsl:when>
</xsl:choose>
<!-- some data from transform may not have new line such as Format: -->
<!-- we will treat the whole data as a string -->
<xsl:variable name="outputString">
<xsl:choose>
<xsl:when test="contains($data, $new_line)">
<xsl:value-of select="substring-before($data, $new_line! e)"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$data"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:variable name="outputIdx">
<xsl:call-template name="removeLeadingTabsSpaces">
<xsl:with-param name="txt" select="$outputString"/>
<xsl:with-param name="data" select="$outputString! g"/>
<!-- ;/xsl:call-template>
</xsl:variable>
<!-- need further process the string if it is longer than the column width -->
<xsl:variable name="displayedString">
<xsl:choose>
<xsl:when test="$outputIdx > 0">
<xsl:value-of select="substring($outputString, $outputIdx, string-length($outputString) - $outputIdx + 1)"/>
</xsl:when>
</xsl:choose>
</xsl:variable>
<xsl:choose>
<xsl:when test="string-length($displayedString) > $dataColumnWidth">
<xsl:call-template name="outDisplayedString">
<xsl:with-param name="fullString" select="$displayedString"/>
<xsl:with-param name="curDisplayString" select="substring($displayedString, 1, $dataColumnWidth)"/>
<xsl:with-param name="columnWidth" select="$dataColumnWidth"/>
<xsl:with-param name="labelColumnWidth" select="$labelColumnWidth"/>
<xsl:with-param name="padLabel" select="no&! #39;"/>
<xsl:with-param name="curIdx" select="1"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$displayedString"/>
</xsl:choose>
<xsl:when test="$dataColumnWidth > string-length($displayedString)">
<xsl:call-template name="fillSpace" >
<xsl:with-param name="cnt" select="$dataColumnWidth - string-length($displayedString)"/>
<xsl:with-param name="chr" select="" />

```

```

</xsl:call-template>
</xsl:when>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
<xsl:value-of select="$new_line"/>
<xsl:variable name="remainingString">
<xsl:value-of select="substring-after($data, $new_line)"/>
</xsl:variable>
<!-- check if we have second line and update variables, call same templ! e -->
<xsl:if test=" string-length($remainingString) > 0">
<xsl:variable name="updatedHasLabel">
<xsl:choose>
<xsl:when test="$hasLabel = 'yes'">
<xsl:value-of select="no"/>
</xsl:when>
<xsl:otherwise>
<!-- unchanged -->
<xsl:value-of select="$hasLabel"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:call-template name="outputData">
<xsl:with-param name="data" select="$remainingString"/>
<xsl:with-param name="hasLabel" select="$updatedHasLabel"/>
<xsl:with-param name="labelColumnWidth" select="$labelColumnWidth"/>
<xsl:with-param name="dataColumnWidth" select="$dataColumnWidth"/>
</xsl:call-template>
&lt; /xsl:if>
&lt; /xsl:template>

<xsl:template name="outputLabelAndData">
<xsl:param name="displayData"/>
<xsl:param name="labelColumnWidth"/>
<xsl:param name="dataColumnWidth"/>

<xsl:variable name="printedLabel">
<xsl:call-template name="getOutputLabel" />
</xsl:variable>
<xsl:choose>
<!-- if longer than the label column width! , print it and add a new line -->
<xsl:when test="string-length($printedLabel)">
<xsl:choose>
<xsl:when test="string-length($printedLabel) > $labelColumnWidth">
<xsl:value-of select="$printedLabel" />
<xsl:value-of select="$new_line"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$printedLabel" />
<xsl:call-template name="fillSpace" >
<xsl:with-param name="cnt" select="$labelColumnWidth - string-length($printedLabel)"/>
<xsl:with-param name="chr" select=" " />
</xsl:call-template>

```



```

</xsl:otherwise>
</xsl:choose>
</xsl:when>
</xsl:choose>
<!-- only output data if there is data -->
<xsl:choose>
<xsl:when test="$displayData">
<!-- if there is a short label on the same line set haveLabel to true -->
<!-- to print out data on the same line, just like classic webvoyage -->
<xsl:variable name="hasLabel">
<xsl:choose>
<xsl:when test="string-length($printedLabel)">
<xsl:choose>
<xsl:when test="string-length($printedLabel) &lt;= $labelColumnWidth">
<xsl:value-of select="yes"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="no"/>!
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="no"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>

<xsl:call-template name="outputData">
<xsl:with-param name="data" select="$displayData"/>
<xsl:with-param name="hasLabel" select="$hasLabel"/>
<xsl:with-param name="labelColumnWidth" select="$labelColumnWidth" />
<xsl:with-param name="dataColumnWidth" select="$dataColumnWidth" />
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$new_line"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- Holding data from the transformer is a group of carriage return separated -->
<!-- label and data pair and should not break them apart -->
<xsl:template name="outputHoldingFieldData">
<xsl:param name="holdingDisplayData"/>

<xsl:value-of select="substring-before($holdingDisplayData, $new_line)"/>
<xsl:value-of select="$new_line"/>
<xsl:variable name="remaining">
<xsl:value-of select="substring-after($holdingDisplayData, $new_line)"/>
</xsl:variable>
<xsl:choose>
<xsl:when test="string-length($remaining)">
<xsl:call-template name="outputHoldingFieldData">

```

```

<xsl:with-param name="holdingDisplayData" select="$remaining"/>
</xsl:call-template>
</xsl:when>
</xsl:choose>
</xsl:template>

<xsl:template name="outputDisplayData">
<xsl:param name="displayData"/>

<!-- Classic webvoyage, label column 20 characters, data column 58 characters -->
<xsl:variable name="labelColumnWidth">
<xsl:value-of select="20"/>
</xsl:variable>
<xsl:variable name="dataColumnWidth">
<xsl:value-of select="58"/>
</xsl:variable>

<xsl:choose>
<xsl:when test="string-length($displayData) and displayTag/@field='9000'">
<xsl:choose>
<xsl:when test="string-length(@label)">
<xsl:value-of select="@label"/>
</xsl:when>
</xsl:choose>

<xsl:value-of select="$new_line"/>
<!-- holding data is built as a group of carriage return -separated -->
<!-- subfields, need to be handled differently for output -->
<xsl:call-template name="outputHoldingFieldData" >
<xsl:with-param name="holdingDisplayData" select="$displayData"/>
</xsl:call-template>
</xsl:when>

<xsl:when test="string-length($displayData)">
<xsl:call-template name="outputLabelAndData">
<xsl:with-param name="displayData" select="$displayData"/>
<xsl:with-param name="labelColumnWidth" select="$labelColumnWidth"/>
<xsl:with-param name="dataColumnWidth" select="$dataColumnWidth"/>
</xsl:call-template>
<!-- :xsl:when>

<xsl:when test="not(string-length($displayData))">
<xsl:if test="string-length(@notFound)">
<xsl:call-template name="outputLabelAndData">
<xsl:with-param name="displayData" select="@notFound"/>
<xsl:with-param name="labelColumnWidth" select="$labelColumnWidth"/>
<xsl:with-param name="dataColumnWidth" select="$dataColumnWidth"/>
</xsl:call-template>
</xsl:if>
</xsl:when>

```

Original lines 554 - 560

```
</xsl:choose>
</xsl:for-each>
-->
<xsl:for-each select="slim:subfield">
<xsl:choose>
<xsl:when test="contains($subfield,@code)">
```

Original lines 561 - 571

```
<xsl:value-of select="@code"/>
</xsl:variable>
<xsl:variable name="subfieldPreText">
<xsl:if test="$displayTag"><xsl:value-of
select="$displayTag/subfield[@value=$subfieldCode]/@preText"/></xsl:if>
</xsl:variable>
<xsl:variable name="subfieldPostText">
<xsl:if test="$displayTag"><xsl:value-of
select="$displayTag/subfield[@value=$subfieldCode]/@postText"/></xsl:if>
</xsl:variable>

<xsl:choose>
```

```
</xsl:choose>
```

New lines 892 - 900

```
</xsl:choose>
</xsl:for-each>
-->
<!-- output the prefix and postfix only once for this group of subfields for this tag -->
<!-- the prefix and post fix are defined in endNoteCitationConfig.xml for 240 field -->
<xsl:for-each select="slim:subfield">
<xsl:choose>
<xsl:when test="contains($subfield,@code)">
```

New lines 901 - 911

```
<xsl:value-of select="@code"/>
</xsl:variable>
<xsl:variable name="subfieldPreText">
<xsl:if test="($displayTag) and (position() = 1)"><xsl:value-of select="$displayTag/@preText"/></xsl:if>
</xsl:variable>
<xsl:variable name="subfieldPostText">
<xsl:if test="($displayTag) and (position() = last())"><xsl:value-of select="$displayTag/@postText"/></xsl:if>
</xsl:variable>

<xsl:choose>
```

Changed File tomcat/vwebv/context/vwebv/ui/exl_default/xsl/userTextConfigs/limits.xml

Original lines 15 - 23

```
** Version : 8.2.0
** Created : 05-DEC-2007
** Orig Author : David Sellers
** Last Modified : 09-APR-2012
** Last Modified By : Mel Pemble
-->
```

```
<pageLimits>
```

Original lines 60 - 66

```
<limitGroup type="language" label="Language:">
<!--Updated per http://www.loc.gov/marc/languages/language\_code.html 23 January 2015 MD -->

<limit code="all" default="Y">All Languages</limit>
<limit code="ENG">English</limit>
```

Original lines 355 - 361

```
<limit code="MDF">Moksha</limit>
<limit code="MON">Mongolian</limit>
<limit code="LOL">Mongo-Nkundu</limit>
```

New lines 15 - 23

```
** Version : 8.2.0
** Created : 05-DEC-2007
** Orig Author : David Sellers
** Last Modified : 17-JAN-2018
** Last Modified By : Mark Gobat
-->
```

```
<pageLimits>
```

New lines 60 - 67

```
<limitGroup type="language" label="Language:">
<!--Updated per http://www.loc.gov/marc/languages/language\_code.html 23 January 2015 MD -->
<!--Updated per http://www.loc.gov/marc/languages/language\_code.html 17 January 2018 MD -->

<limit code="all" default="Y">All Languages</limit>
<limit code="ENG">English</limit>
```

New lines 356 - 363

```
<limit code="MDF">Moksha</limit>
<limit code="MON">Mongolian</limit>
<limit code="LOL">Mongo-Nkundu</limit>
```

```
<limit code="MKH">Mon-Khmer (Other)</limit>
<limit code="MOS">MoorÃ©</limit>
<limit code="MUL">Multiple languages</limit>
```

```
<limit code="CNR">Montenegrin</limit>
<limit code="MKH">Mon-Khmer (Other)</limit>
<limit code="MOS">MoorÃ©</limit>
<limit code="MUL">Multiple languages</li! mit>
```

Changed File tomcat/vwebv/context/vwebv/ui/exl_default/xsl/userTextConfigs/pageProperties.xml

Original lines 440 - 446

```
<br/>You have more than the maximum allowed demerits for consortial borrowing.
</pageMsg>
<pageMsg errorCode="searchResults.noHits">Search resulted in no hits.</pageMsg>
</pageMessages>
```

New lines 440 - 447

```
<br/>You have more than the maximum allowed demerits for consortial borrowing.
</pageMsg>
<pageMsg errorCode="searchResults.noHits">Search resulted in no hits.</pageMsg>
<pageMsg errorCode="searchResults.truncated">Search results were truncated.</pageMsg>
</pageMessages>
```