



This document describes the files in the skin directories changed from 8.2.1 to 8.2.2 as well as what has been changed within the skin files. It is meant to be used when planning for the best way to carry your skin-level customizations from a previous version of WebVoyage forward to version 8.2.2. If you have not altered a file, you can copy the new 8.2.2 version over the existing file in whatever skin you would like. If you have customized a file, we recommend that you review the changes below but recommit your customizations to the *new* file. Test your skin after each customization is reapplied to the new files.

If you are upgrading from a version of Voyager prior to 8.0.0, please note that there is not a Changed Skin Files document. Since the process for upgrading to Voyager 8.0.0 and higher from an earlier release requires that you begin with the Voyager 8.x version of all skin files, no list of changed files is necessary.

Please see the Voyager 8.2.2 Release Notes for specific bug fixes that affect skin files. The Release Notes can be found on the Documentation Center under
Home > Documentation Center > Ex Libris Documentation Center > Voyager > Upgrade Release Planning > 8.2.2.

Changed skin files from 8.2.1 to 8.2.2

The following skin files changed in 8.2.2.

The files below are all in the directory, /m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/ (e.g. the first one is /m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/css/displayCommon.css):

Changed Files:

/css/displayCommon.css
/webvoyage.properties
/webvoyage.properties.tml
/xsl/contentLayout/cl_myAccount.xsl
/xsl/contentLayout/cl_searchMySearches.xsl
/xsl/contentLayout/configs/102X_display.xsl
/xsl/contentLayout/configs/104X_display.xsl
/xsl/contentLayout/configs/displaycfg.xml
/xsl/contentLayout/display/display.xsl
/xsl/printFormats/printCommon.xsl

The upgrade to 8.2.2 will overwrite your exl_default directory with all of the new files listed above, as well as the default versions of all other WebVoyage skin files.

The following vxws files changed in 8.2.2.

The files below are all in the directory, /m1/voyager/xxxdb/tomcat/vxws/context/vxws (e.g. the first one is /m1/voyager/xxxdb/tomcat/vxws/context/vxws/ui/exl_default/vxws.properties):

/ui/exl_default/vxws.properties
/ui/exl_default/vxws.properties.tml
/xsl/configs/104X_display.xsl
/xsl/configs/OPAC_in_primo_104X_display.xsl
/xsl/transformIndexHoldings.xsl

/xsl/transformOPACInPrimoHoldings.xsl

Additionally, Voyager 8.2.2 contains the following new files in the tomcat directory (/m1/voyager/xxxdb/tomcat):

New Files:

/vpds/context/vpds/error_404.html
/vwebv/context/vwebv/error_404.html
/vwebv/context/vwebv/error_404.html.tml
/vxws/context/vxws/error_404.html
/vxws/context/vxws/error_500.html
/vxws/context/vxws/error_503.html
/vxws/context/vxws/xsl/transformIndexHoldings.xsl
/vxws/context/vxws/xsl/transformSupplementHoldings.xsl

Skin file content changes 8.2.1 to 8.2.2

Below contains exactly what has been changed in each file for 8.2.2.

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/css/displayCommon.css

Action: Change lines 402 to 403 to the following snippet

```
+++++++ Snippet Start +++++++  
margin:0px;  
min-width:400px;  
+++++++ Snippet End +++++++
```

Action: Add the following snippet at starting at line 398

```
+++++++ Snippet Start +++++++  
.frameWorkUI #pageContainer  
{  
  min-width: 925px;  
}  
+++++++ Snippet End +++++++
```

Action: Change line 165 to the following snippet

```
+++++++ Snippet Start +++++++  
min-width: 240px;  
+++++++ Snippet End +++++++
```

Action: Change line 1 to the following snippet

```
+++++++ Snippet Start +++++++  
/* $Revision: 1.6.4.1.2.1 $ $Date: 2013/05/01 21:44:58 $ */  
+++++++ Snippet End +++++++
```

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/webvoyage.properties

Action: Change line 2210 to the following snippet

```
+++++++ Snippet Start +++++++  
#####imageServer.scanDoc=__WEBVOYAGE_CLASSIC_URL___/cgi-bin/Pscandoc.cgi?  
#imageServer.scanDoc=__SCANDOC_CGI_URL___  
+++++++ Snippet End +++++++
```

Action: Add the following snippet at starting at line 1576

```
+++++++ Snippet Start +++++++  
page.myAccount.chargedItem.dueDate.indef=Indefinite  
+++++++ Snippet End +++++++
```

```
#### Action: Change line 960 to the following snippet ####
+++++++ Snippet Start ++++++++
# for the related items link when the maxPerPage value is met.
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 1 to the following snippet ####
+++++++ Snippet Start ++++++++
# $Revision: 1.23.2.3.2.2.2.3 $ $Date: 2013/05/01 21:44:58 $
+++++++ Snippet End ++++++++
```

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/webvoyage.properties.tml

```
#### Action: Change line 2210 to the following snippet ####
+++++++ Snippet Start ++++++++
#####imageServer.scanDoc=__WEBVOYAGE_CLASSIC_URL___/cgi-bin/Pscandoc.cgi?
#imageServer.scanDoc=__SCANDOC_CGI_URL___
+++++++ Snippet End ++++++++
```

```
#### Action: Add the following snippet at starting at line 1576 ####
+++++++ Snippet Start ++++++++
page.myAccount.chargedItem.dueDate.indef=Indefinite
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 960 to the following snippet ####
+++++++ Snippet Start ++++++++
# for the related items link when the maxPerPage value is met.
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 1 to the following snippet ####
+++++++ Snippet Start ++++++++
# $Revision: 1.23.2.3.2.2.2.3 $ $Date: 2013/05/01 21:44:58 $
+++++++ Snippet End ++++++++
```

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/cl_myAccount.xsl

```
#### Action: Change lines 964 to 965 to the following snippet ####
+++++++ Snippet Start ++++++++
                <label for="bookItemId_{position()}_page:itemId">
                    <input type="checkbox" name="selectBookings"
id="bookItemId_{position()}_page:itemId" value="{page:itemId}" />
+++++++ Snippet End ++++++++
```

```
#### Action: Change lines 720 to 721 to the following snippet ####
+++++++ Snippet Start ++++++++
                <label for="pendlItemId_{position()}_page:itemId">
                    <input type="checkbox" name="selectRequest"
id="pendlItemId_{position()}_page:itemId" value="{page:itemId}" />
+++++++ Snippet End ++++++++
```

```
#### Action: Change lines 555 to 556 to the following snippet ####
+++++++ Snippet Start ++++++++
                <label for="availReqItemId_{position()}_page:itemId">
                    <input type="checkbox" name="selectItem"
id="availReqItemId_{position()}_page:itemId" value="{page:itemId}" />
+++++++ Snippet End ++++++++
```

```
#### Action: Change lines 399 to 400 to the following snippet ####
+++++++ Snippet Start ++++++++
                <label for="chargedItemId_{position()}_page:itemId">
                <input type="checkbox" id="chargedItemId_{position()}_page:itemId"
name="selectCharged" value="{page:itemId}" />
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 18 to the following snippet ####
+++++++ Snippet Start ++++++++
**    Last Modified : 11-APR-2013
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 15 to the following snippet ####
+++++++ Snippet Start ++++++++
**    Version : 8.2.2
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 2 to the following snippet ####
+++++++ Snippet Start ++++++++
<!-- $Revision: 1.5.4.2.2.1 $ $Date: 2013/04/11 17:02:46 $ -->
+++++++ Snippet End ++++++++
```

[/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/cl_searchMySearches.xsl](#)

```
#### Action: Change line 15 to the following snippet ####
+++++++ Snippet Start ++++++++
**    Version : 8.2.2
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 2 to the following snippet ####
+++++++ Snippet Start ++++++++
<!-- $Revision: 1.4.4.2.2.1 $ $Date: 2013/03/27 15:11:52 $ -->
+++++++ Snippet End ++++++++
```

[/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/configs/102X_display.xsl](#)

```
#### Action: Change line 151 to the following snippet ####
+++++++ Snippet Start ++++++++

+++++++ Snippet End ++++++++
```

```
#### Action: Change line 109 to the following snippet ####
+++++++ Snippet Start ++++++++
                <xsl:if test="position() > $totalNumberOfReceivedIssues -
$numberOfIssuesToDisplayPerComponent">
+++++++ Snippet End ++++++++
```

```
#### Action: Change line 102 to the following snippet ####
+++++++ Snippet Start ++++++++
                <!-- </xsl:if> -->
+++++++ Snippet End ++++++++
```

```
#### Action: Add the following snippet at starting at line 98 ####
+++++++ Snippet Start ++++++++
```

-->
+++++++ Snippet End ++++++

Action: Add the following snippet at starting at line 97

+++++++ Snippet Start ++++++
 <!-- This parameter should not apply to un-received issues
+++++++ Snippet End ++++++

Action: Add the following snippet at starting at line 83

+++++++ Snippet Start ++++++

 <xsl:variable name="totalNumberOfReceivedIssues">
 <xsl:value-of select="count(\$component/mfhd:receivedIssues/mfhd:receivedIssue)"/>
 </xsl:variable>

+++++++ Snippet End ++++++

Action: Change line 2 to the following snippet

+++++++ Snippet Start ++++++
<!-- \$Revision: 1.5.4.2.2.1 \$ \$Date: 2013/05/13 18:57:05 \$ -->
+++++++ Snippet End ++++++

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/configs/104X_display.xsl

Action: Add the following snippet at starting at line 1527

+++++++ Snippet Start ++++++
<!-- ***** -->

```
<xsl:template name="countHyphenInSubfield">
  <xsl:param name="subfield"/>

  <xsl:variable name="hyphenCounter">
    <!-- Remove hyphens from string and check the difference to get the counter. -->
    <xsl:value-of select="string-length($subfield) - string-length(translate($subfield, '-', ''))"/>
  </xsl:variable>
  <xsl:value-of select="$hyphenCounter"/>
</xsl:template>
```

<!-- ***** -->

```
<xsl:template name="getTextAfterSecondHyphen">
  <xsl:param name="text"/>

  <xsl:variable name="textAfterHyphen">
    <xsl:value-of select="substring-after($text, '-')"/>
  </xsl:variable>

  <xsl:value-of select="substring-after($textAfterHyphen, '-')"/>
</xsl:template>
```

+++++++ Snippet End ++++++

Action: Change line 1350 to the following snippet

+++++++ Snippet Start ++++++
 <xsl:when test="\$hyphenCount = 1">
+++++++ Snippet End ++++++

Action: Add the following snippet at starting at line 1348

+++++++ Snippet Start +++++++

+++++++ Snippet End +++++++

Action: Change line 1346 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:when test="$hyphenCount = 1 and substring-after($text, '-')">
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 1344

+++++++ Snippet Start +++++++

```
<!-- Expecting three hyphens if there is more than 1 -->
```

```
<xsl:when test="$hyphenCount > 1">
```

```
  <xsl:variable name="textAfterSecondHyphen">
```

```
    <xsl:call-template name="getTextAfterSecondHyphen">
```

```
      <xsl:with-param name="text" select="$text"/>
```

```
    </xsl:call-template>
```

```
  </xsl:variable>
```

```
<!-- Add a hyphen before the end part -->
```

```
<xsl:variable name="textAfterSecondHyphenWithPreHyphen">
```

```
  <xsl:value-of select="concat('-', $textAfterSecondHyphen)"/>
```

```
</xsl:variable>
```

```
<xsl:value-of select="substring-before($text, $textAfterSecondHyphenWithPreHyphen)"/>
```

```
</xsl:when>
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 1343

+++++++ Snippet Start +++++++

```
<xsl:variable name="hyphenCount">
```

```
  <xsl:call-template name="countHyphenInSubfield">
```

```
    <xsl:with-param name="subfield" select="$text"/>
```

```
  </xsl:call-template>
```

```
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change line 1210 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($display)">
```

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

```
</xsl:if>
```

+++++++ Snippet End +++++++

Action: Change lines 1203 to 1208 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:choose>
```

```
  <xsl:when test="$hyphenCount = '1' or $hyphenCount != '3'">
```

```
    <xsl:call-template name="getChron">
```

```
      <xsl:with-param name="caption" select="$subfield5j"/>
```

```
      <xsl:with-param name="text" select="substring-after($subfield6j, '-')"/>
```

```
      <xsl:with-param name="lang" select="$lang"/>
```

```
      <xsl:with-param name="frequency" select="$subfield5w"/>
```

```
    </xsl:call-template>
```

```
  </xsl:when>
```

```
<xsl:otherwise>
```

```
  <xsl:variable name="textAfterSecondHyphen">
```

```
    <xsl:call-template name="getTextAfterSecondHyphen">
```

```

    <xsl:with-param name="text" select="$subfield6j"/>
  </xsl:call-template>
</xsl:variable>

<xsl:call-template name="getChron">
  <xsl:with-param name="caption" select="$subfield5j"/>
  <xsl:with-param name="text" select="$textAfterSecondHyphen"/>
  <xsl:with-param name="lang" select="$lang"/>
  <xsl:with-param name="frequency" select="$subfield5w"/>
</xsl:call-template>

```

```

</xsl:otherwise>
</xsl:choose>

```

+++++++ Snippet End ++++++

Action: Add the following snippet at starting at line 1201

+++++++ Snippet Start ++++++

```

<!-- Count hyphens. Multiple dashes indicate a quarterly issue chronology -->
<xsl:variable name="hyphenCount">
  <xsl:call-template name="countHyphenInSubfield">
    <xsl:with-param name="subfield" select="$subfield6j"/>
  </xsl:call-template>
</xsl:variable>

```

+++++++ Snippet End ++++++

Action: Change lines 1186 to 1194 to the following snippet

+++++++ Snippet Start ++++++

```

<xsl:variable name="displayl">
  <xsl:choose>
    <xsl:when test="string-length($subfield5i) and contains($subfield6i, '-')">
      <xsl:call-template name="getChron">
        <xsl:with-param name="caption" select="$subfield5i"/>
        <xsl:with-param name="text" select="substring-after($subfield6i, '-')"/>
        <xsl:with-param name="lang" select="$lang"/>
        <xsl:with-param name="frequency" select="0"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <!-- Need some value to test for chronology level separator -->
      <xsl:value-of select = """/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

```

```

  <xsl:if test="string-length($displayl)">

```

+++++++ Snippet End ++++++

Action: Add the following snippet at starting at line 1112

+++++++ Snippet Start ++++++

+++++++ Snippet End ++++++

Action: Delete line 823

Action: Change line 2 to the following snippet

+++++++ Snippet Start ++++++

```

<!-- $Revision: 1.6.4.2.2.1 $ $Date: 2013/05/10 15:27:46 $ -->

```

+++++++ Snippet End ++++++

[/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/configs/displaycfg.xml](#)

Action: Add the following snippet at starting at line 274

+++++++ Snippet Start +++++++

```
<displayTags label="Preceding Titles:" notFound="No preceding titles found.">
  <displayTag field="3500" profileMatch="ISSNPREC"/>
</displayTags>
```

```
<displayTags label="Succeeding Titles:" notFound="No succeeding titles found.">
  <displayTag field="3500" profileMatch="ISSNSUCC"/>
</displayTags>
```

+++++++ Snippet End +++++++

Action: Change lines 271 to 272 to the following snippet

+++++++ Snippet Start +++++++

```
<!-- These profiles should match the profiles set in the webvoyage.properties as well as setup in sysadmin -->
```

```
<displayTags label="Related Records">
  <displayTag field="3500" profileMatch="ISSNCOMP"/>
```

+++++++ Snippet End +++++++

Action: Change line 2 to the following snippet

+++++++ Snippet Start +++++++

```
<!-- $Revision: 1.5.2.1.2.1.2.1 $ $Date: 2013/05/01 21:44:58 $ -->
```

+++++++ Snippet End +++++++

[/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/contentLayout/display/display.xsl](#)

Action: Add the following snippet at starting at line 1726

+++++++ Snippet Start +++++++

```
<xsl:for-each select="page:element[@nameId=$profileName]">
```

+++++++ Snippet End +++++++

Action: Delete line 1725

Action: Add the following snippet at starting at line 1722

+++++++ Snippet Start +++++++

```
<xsl:variable name="profileName">
  <xsl:value-of select="'holdingsInfo.relatedRecords.profile.'"/><xsl:value-of select="@profileMatch"/>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change line 971 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:value-of select="$searchParameter"/><xsl:value-of
select="$searchArgParameter"/>=<xsl:value-of select="normalize-
space(substring($cleanedRedirectSearchData,$redirectNonFilingIndicator))"/>&amp;<xsl:value-of
select="$searchCodeParameter"/>=<xsl:value-of select="$cleanedRedirectSearchCode"/>&amp;<xsl:value-of
select="$searchTypeParameter"/>=<xsl:value-of select="$redirectSearchType"/>
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 968

+++++++ Snippet Start +++++++

```
<xsl:variable name="cleanedRedirectSearchCode">
  <xsl:call-template name="url-encode">
    <xsl:with-param name="str" select="$redirectSearchCode"/>
  </xsl:call-template>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change line 962 to the following snippet

+++++++ Snippet Start +++++++

+++++++ Snippet End +++++++

Action: Change line 953 to the following snippet

+++++++ Snippet Start +++++++

+++++++ Snippet End +++++++

Action: Change line 838 to the following snippet

+++++++ Snippet Start +++++++

+++++++ Snippet End +++++++

Action: Change line 829 to the following snippet

+++++++ Snippet Start +++++++

```
<!-- For all searches, replace the pound (#) with the URL encoding %23.
```

+++++++ Snippet End +++++++

Action: Change line 819 to the following snippet

+++++++ Snippet Start +++++++

+++++++ Snippet End +++++++

Action: Change line 98 to the following snippet

+++++++ Snippet Start +++++++

```
<!-- ## The following variables are used to convert case ##
```

+++++++ Snippet End +++++++

Action: Change line 2 to the following snippet

+++++++ Snippet Start +++++++

```
<!-- $Revision: 1.17.2.1.2.1.2.2 $ $Date: 2013/05/01 21:44:57 $ -->
```

+++++++ Snippet End +++++++

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/exl_default/xsl/printFormats/printCommon.xsl

Action: Change line 61 to the following snippet

+++++++ Snippet Start +++++++

```
</img>
<!-- ## Display the Page Heading ## -->
<h2>
  <xsl:call-template name="buildPageHeading">
    <xsl:with-param name="nameId" select="page.heading"/>
  </xsl:call-template>
</h2>
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 36

+++++++ Snippet Start +++++++

```
<!-- ## All pages need a title ## -->
<title>
  <xsl:call-template name="buildPageTitle">
    <xsl:with-param name="nameId" select="page.title"/>
  </xsl:call-template>
</title>
```

+++++++ Snippet End +++++++

Action: Change line 17 to the following snippet

+++++++ Snippet Start +++++++

** Last Modified : 08-APR-2013

+++++++ Snippet End +++++++

Action: Change line 14 to the following snippet

+++++++ Snippet Start +++++++

** Version : 8.2.2

+++++++ Snippet End +++++++

Action: Change line 2 to the following snippet

+++++++ Snippet Start +++++++

<!-- \$Revision: 1.5.4.1.2.1 \$ \$Date: 2013/04/08 21:58:37 \$ -->

+++++++ Snippet End +++++++

/m1/voyager/xxxdb/tomcat/vxws/context/vxws/ui/exl_default/vxws.properties

Action: Add the following snippet at starting at line 214

+++++++ Snippet Start +++++++

#####

Indicate whether or not you want the various serial issues to be
sent to Primo via OvP.

#

#####

```
serial.issue.1020.recent.basic=Y
serial.issue.1022.recent.supplement=Y
serial.issue.1024.recent.index=Y
serial.issue.1040.older.basic=Y
serial.issue.1042.older.supplement=Y
serial.issue.1044.older.index=Y
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 202

+++++++ Snippet Start +++++++

page.myAccount.chargedItem.dueDate.indef=Indefinite

+++++++ Snippet End +++++++

/m1/voyager/xxxdb/tomcat/vxws/context/vxws/ui/exl_default/vxws.properties.tml

Action: Add the following snippet at starting at line 214

+++++++ Snippet Start +++++++

#####

Indicate whether or not you want the various serial issues to be
sent to Primo via OvP.

```
#
#####
serial.issue.1020.recent.basic=Y
serial.issue.1022.recent.supplement=Y
serial.issue.1024.recent.index=Y
serial.issue.1040.older.basic=Y
serial.issue.1042.older.supplement=Y
serial.issue.1044.older.index=Y
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 202

```
+++++++ Snippet Start ++++++++
page.myAccount.chargedItem.dueDate.indef=Indefinite
+++++++ Snippet End ++++++++
```

/m1/voyager/xxxdb/tomcat/vxws/context/vxws/xsl/configs/104X_display.xsl

Action: Add the following snippet at starting at line 1467

```
+++++++ Snippet Start ++++++++
<xsl:template name="countHyphenInSubfield">
  <xsl:param name="subfield"/>

  <xsl:variable name="hyphenCounter">
    <!-- Remove hyphens from string and check the difference to get the counter. -->
    <xsl:value-of select="string-length($subfield) - string-length(translate($subfield, '-', ''))"/>
  </xsl:variable>
  <xsl:value-of select="$hyphenCounter"/>
</xsl:template>
```

```
<!-- ***** -->
```

```
<xsl:template name="getTextAfterSecondHyphen">
  <xsl:param name="text"/>

  <xsl:variable name="textAfterHyphen">
    <xsl:value-of select="substring-after($text, '-')"/>
  </xsl:variable>

  <xsl:value-of select="substring-after($textAfterHyphen, '-')"/>
</xsl:template>
```

+++++++ Snippet End ++++++++

Action: Add the following snippet at starting at line 1466

```
+++++++ Snippet Start ++++++++
<!-- ***** -->
+++++++ Snippet End ++++++++
```

Action: Change lines 1455 to 1465 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:param name="caption"/>
  <xsl:choose>
    <xsl:when test="(starts-with($caption, '(') and (substring($caption, string-length($caption)) = '))">
      </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$caption"/>
      <xsl:if test="substring($caption, (string-length($caption) - string-length($chronLabelDataSeparator)) + 1)
```

```

!= $chronLabelDataSeparator">
    <xsl:value-of select="$chronLabelDataSeparator"/>
  </xsl:if>
</xsl:otherwise>
</xsl:choose>
+++++++ Snippet End ++++++

#### Action: Change lines 1447 to 1449 to the following snippet ####
+++++++ Snippet Start ++++++
  <xsl:param name="text"/>
  <xsl:variable name="upperCaseText" select="translate($text, $lowercase, $uppercase)"/>
  <xsl:value-of select="$upperCaseText"/>
+++++++ Snippet End ++++++

#### Action: Change lines 1438 to 1442 to the following snippet ####
+++++++ Snippet Start ++++++
  <!-- if no value was matched for chronCode, display the given chronCode -->
  <xsl:choose>
    <xsl:when test="string-length($text)"><xsl:value-of select="$text"/></xsl:when>
    <xsl:otherwise><xsl:value-of select="$chronCode"/></xsl:otherwise>
  </xsl:choose>
+++++++ Snippet End ++++++

#### Action: Change lines 1435 to 1436 to the following snippet ####
+++++++ Snippet Start ++++++
  <!-- get the value for chronCode -->
  <xsl:variable name="text" select="$chronValues//lang[@id=$language]/chron[@id=$chroncodeNoZero]"/>
+++++++ Snippet End ++++++

#### Action: Change lines 1423 to 1433 to the following snippet ####
+++++++ Snippet Start ++++++
  <!-- chronCode may or may not have a leading zero - so let's strip any leading zero's -->
  <xsl:variable name="chroncodeNoZero">
    <xsl:choose>
      <xsl:when test="starts-with($chronCode, '0')">
        <xsl:value-of select="substring-after($chronCode, '0')"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$chronCode"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
+++++++ Snippet End ++++++

#### Action: Change lines 1415 to 1421 to the following snippet ####
+++++++ Snippet Start ++++++
  <!-- if chronLangUC was found, use that for language. Otherwise, default to ENG -->
  <xsl:variable name="language">
    <xsl:choose>
      <xsl:when test="string-length($testForLang)"><xsl:value-of select="$testForLang"/></xsl:when>
      <xsl:otherwise>ENG</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
+++++++ Snippet End ++++++

#### Action: Change lines 1412 to 1413 to the following snippet ####
+++++++ Snippet Start ++++++
  <!-- see if <lang> element with chronLangUC attribute exists in chronValues file -->
  <xsl:variable name="testForLang"><xsl:value-of
select="$chronValues//lang[@id=$chronLangUC]/@id"/></xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 1405 to 1410 to the following snippet

+++++++ Snippet Start +++++++

```
<!-- uppercase chronLang -->
<xsl:variable name="chronLangUC">
  <xsl:call-template name="upperCase">
    <xsl:with-param name="text" select="$chronLang"/>
  </xsl:call-template>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 1402 to 1403 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:param name="chronCode"/>
<xsl:param name="chronLang"/>
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 1396

+++++++ Snippet Start +++++++

```
<xsl:otherwise>
  <!-- no chron code - just display text -->
  <xsl:value-of select="$text"/>
</xsl:otherwise>
</xsl:choose>
```

+++++++ Snippet End +++++++

Action: Change lines 1391 to 1395 to the following snippet

+++++++ Snippet Start +++++++

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

+++++++ Snippet End +++++++

Action: Change lines 1388 to 1389 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:value-of select="$temp1Name"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 1383 to 1386 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:otherwise>
  <xsl:variable name="temp1Name">
    <xsl:call-template name="translateChron">
      <xsl:with-param name="chronCode" select="$text"/>
      <xsl:with-param name="chronLang" select="$lang"/>
    </xsl:call-template>
  </xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 1375 to 1381 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:value-of select="concat($temp1aName, '/', $temp1bName)"/>
</xsl:when>
```

+++++++ Snippet End +++++++

Action: Change lines 1372 to 1373 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="temp1bName">
```

```

        <xsl:call-template name="translateChron">
          <xsl:with-param name="chronCode" select="$temp1b"/>
          <xsl:with-param name="chronLang" select="$lang"/>
        </xsl:call-template>
      </xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 1365 to 1370 to the following snippet

```

+++++++ Snippet Start ++++++
      <xsl:variable name="temp1aName">
        <xsl:call-template name="translateChron">
          <xsl:with-param name="chronCode" select="$temp1a"/>
          <xsl:with-param name="chronLang" select="$lang"/>
        </xsl:call-template>
      </xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 1358 to 1363 to the following snippet

```

+++++++ Snippet Start ++++++
  <!-- do we need to translate a chronology code? -->
  <xsl:choose>
    <xsl:when test="contains($caption, 'month') or contains($caption, 'season')">
      <!-- yes -->
      <xsl:variable name="chronName">
        <xsl:choose>
          <!-- check for double chrons -->
          <xsl:when test="contains($text, '/')">
            <xsl:variable name="temp1a" select="substring-before($text, '/')"/>
            <xsl:variable name="temp1b" select="substring-after($text, '/')"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:variable name="temp1a" select="$text"/>
            <xsl:variable name="temp1b" select=""/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name="chronName" select="$caption"/>
    </xsl:otherwise>
  </xsl:choose>
+++++++ Snippet End ++++++

```

Action: Change lines 1343 to 1356 to the following snippet

```

+++++++ Snippet Start ++++++
  <xsl:param name="caption"/>
  <xsl:param name="text"/>
  <xsl:param name="lang"/>
+++++++ Snippet End ++++++

```

Action: Change lines 1322 to 1337 to the following snippet

```

+++++++ Snippet Start ++++++
  <!--
    if they are the same, don't use anything
    unless some suffix data has already been presented
  </xsl:choose>
  <xsl:when test="$before != $after">
    <xsl:value-of select="$after"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:if test="number($previous) > 0">
      <xsl:value-of select="$after"/>
    </xsl:if>
  </xsl:otherwise>
  </xsl:choose>
  -->
  <xsl:value-of select="$after"/>
</xsl:if>
+++++++ Snippet End ++++++

```

Action: Change lines 1314 to 1320 to the following snippet

```

+++++++ Snippet Start ++++++
  <xsl:if test="contains($text, '-')">

```

```

    <xsl:variable name="before">
      <xsl:value-of select="substring-before($text, '-')"/>
    </xsl:variable>
    <xsl:variable name="after">
      <xsl:value-of select="substring-after($text, '-')"/>
    </xsl:variable>
+++++++ Snippet End ++++++++

#### Action: Change lines 1311 to 1312 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:param name="text"/>
  <xsl:param name="previous" select="0"/>
+++++++ Snippet End ++++++++

#### Action: Change lines 1298 to 1305 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:variable name="hyphenCount">
    <xsl:call-template name="countHyphenInSubfield">
      <xsl:with-param name="subfield" select="$text"/>
    </xsl:call-template>
  </xsl:variable>

  <xsl:choose>
    <!-- Expecting three hyphens if there is more than 1 -->
    <xsl:when test="$hyphenCount > 1">
      <xsl:variable name="textAfterSecondHyphen">
        <xsl:call-template name="getTextAfterSecondHyphen">
          <xsl:with-param name="text" select="$text"/>
        </xsl:call-template>
      </xsl:variable>

      <!-- Add a hyphen before the end part -->
      <xsl:variable name="textAfterSecondHyphenWithPreHyphen">
        <xsl:value-of select="concat('-', $textAfterSecondHyphen)"/>
      </xsl:variable>

      <xsl:value-of select="substring-before($text, $textAfterSecondHyphenWithPreHyphen)"/>
    </xsl:when>

    <!-- Is there a hyphen and something after the hyphen? -->
    <xsl:when test="$hyphenCount = 1 and substring-after($text, '-')">
      <xsl:value-of select="substring-before($text, '-')"/>
    </xsl:when>

    <!-- Is there a hyphen? There is nothing after the hyphen but include hyphen in display. -->
    <xsl:when test="$hyphenCount = 1">
      <xsl:value-of select="concat(substring-before($text, '-'), '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$text"/>
    </xsl:otherwise>
  </xsl:choose>

+++++++ Snippet End ++++++++

#### Action: Change line 1296 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:param name="text"/>
+++++++ Snippet End ++++++++

#### Action: Change lines 1287 to 1289 to the following snippet ####

```

```

+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($altChron)">
    <xsl:value-of select="concat(' ', $altChron, ' )'"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1283 to 1285 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($altEnum)">
    <xsl:value-of select="concat('=', $altEnum)"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1279 to 1281 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($chron2)">
    <xsl:value-of select="concat(' ', $chron2, ' )'"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1275 to 1277 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($enum2)">
    <xsl:value-of select="$enum2"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1271 to 1273 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($chron1) or string-length($chron2)">
    <xsl:value-of select="'-'"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1267 to 1269 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($chron1)">
    <xsl:value-of select="concat(' ', $chron1, ' )'"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1263 to 1265 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($enum1)">
    <xsl:value-of select="$enum1"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1259 to 1261 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($lineCaption)">
    <xsl:value-of select="concat($lineCaption, ':')"/>
  </xsl:if>
+++++++ Snippet End ++++++++

#### Action: Change lines 1251 to 1257 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:param name="enum1"/>
  <xsl:param name="enum2"/>
  <xsl:param name="chron1"/>
  <xsl:param name="chron2"/>

```



```
<xsl:param name="altEnum"/>
<xsl:param name="altChron"/>
<xsl:param name="lineCaption"/>
+++++++ Snippet End +++++++
```

Action: Change lines 1242 to 1244 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:if test="string-length($altChron)">
  <xsl:value-of select="concat(' ', $altChron, ' )' )"/>
</xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1239 to 1240 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:value-of select="""/>
</xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1235 to 1237 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($chron2) and $chron2 != $chron1">
    <xsl:value-of select="concat(' - ', $chron2)"/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1232 to 1233 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($chron1)">
    <xsl:value-of select="concat(' ', $chron1)"/>
+++++++ Snippet End ++++++++
```

Action: Change lines 1228 to 1230 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($altEnum)">
    <xsl:value-of select="concat(' ', $altEnum, ' )' )"/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1224 to 1226 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($enum2) and $enum2 != $enum1">
    <xsl:value-of select="concat(' - ', $enum2)"/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1220 to 1222 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($enum1)">
    <xsl:value-of select="$enum1"/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1216 to 1218 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($lineCaption)">
    <xsl:value-of select="concat($lineCaption, ' : ' )"/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1208 to 1214 to the following snippet

```
+++++++ Snippet Start ++++++++
```

```

<xsl:param name="enum1"/>
<xsl:param name="enum2"/>
<xsl:param name="chron1"/>
<xsl:param name="chron2"/>
<xsl:param name="altEnum"/>
<xsl:param name="altChron"/>
<xsl:param name="lineCaption"/>
+++++++ Snippet End ++++++++

```

Action: Add the following snippet at starting at line 1202

```

+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($displayI)">
    <xsl:value-of select="$chronologyLevelSeparator"/>
  </xsl:if>

  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5j"/>
  </xsl:call-template>
  <xsl:value-of select="$displayJ"/>
</xsl:if>

<xsl:if test="string-length($subfield5k) and contains($subfield6k, '-')">
  <xsl:variable name="displayK">
    <xsl:call-template name="getChron">
      <xsl:with-param name="caption" select="$subfield5k"/>
      <xsl:with-param name="text" select="substring-after($subfield6k, '-')"/>
      <xsl:with-param name="lang" select="$lang"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$chronologyLevelSeparator"/>
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5k"/>
  </xsl:call-template>
  <xsl:value-of select="$displayK"/>
</xsl:if>

<xsl:if test="string-length($subfield5l) and contains($subfield6l, '-')">
  <xsl:variable name="displayL">
    <xsl:call-template name="getChron">
      <xsl:with-param name="caption" select="$subfield5l"/>
      <xsl:with-param name="text" select="substring-after($subfield6l, '-')"/>
      <xsl:with-param name="lang" select="$lang"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$chronologyLevelSeparator"/>
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5l"/>
  </xsl:call-template>
  <xsl:value-of select="$displayL"/>
</xsl:if>

```

+++++++ Snippet End ++++++++

Action: Change lines 1188 to 1201 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:call-template name="getChron">
    <xsl:with-param name="caption" select="$subfield5j"/>
    <xsl:with-param name="text" select="$textAfterSecondHyphen"/>
    <xsl:with-param name="lang" select="$lang"/>
  </xsl:call-template>
</xsl:otherwise>

```

```
    </xsl:choose>
  </xsl:variable>
+++++++ Snippet End +++++++
```

Action: Change lines 1173 to 1186 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($subfield5j) and contains($subfield6j, '-')">
    <!-- Count hyphens. Multiple dashes indicate a quarterly issue chronology -->
    <xsl:variable name="hyphenCount">
      <xsl:call-template name="countHyphenInSubfield">
        <xsl:with-param name="subfield" select="$subfield6j"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name="displayJ">
      <xsl:choose>
        <xsl:when test="$hyphenCount = '1' or $hyphenCount != '3'">
          <xsl:message>We are checking out a quarterly</xsl:message>
          <xsl:call-template name="getChron">
            <xsl:with-param name="caption" select="$subfield5j"/>
            <xsl:with-param name="text" select="substring-after($subfield6j, '-')"/>
            <xsl:with-param name="lang" select="$lang"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:variable name="textAfterSecondHyphen">
            <xsl:call-template name="getTextAfterSecondHyphen">
              <xsl:with-param name="text" select="$subfield6j"/>
            </xsl:call-template>
          </xsl:variable>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1158 to 1171 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($displayl)">
    <xsl:call-template name="displayChronCaption">
      <xsl:with-param name="caption" select="$subfield5i"/>
    </xsl:call-template>
    <xsl:value-of select="$displayl"/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 1143 to 1156 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:variable name="displayl">
    <xsl:choose>
      <xsl:when test="string-length($subfield5i) and contains($subfield6i, '-')">
        <xsl:call-template name="getChron">
          <xsl:with-param name="caption" select="$subfield5i"/>
          <xsl:with-param name="text" select="substring-after($subfield6i, '-')"/>
          <xsl:with-param name="lang" select="$lang"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = """/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
+++++++ Snippet End ++++++++
```

Action: Change lines 1133 to 1141 to the following snippet

```
+++++++ Snippet Start ++++++++
```

```

<xsl:param name="lang"/>
<xsl:param name="subfield5i"/>
<xsl:param name="subfield5j"/>
<xsl:param name="subfield5k"/>
<xsl:param name="subfield5l"/>
<xsl:param name="subfield6i"/>
<xsl:param name="subfield6j"/>
<xsl:param name="subfield6k"/>
<xsl:param name="subfield6l"/>
+++++++ Snippet End ++++++++

```

Action: Change lines 1108 to 1127 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:if test="string-length($subfield5l) and string-length($subfield6l)">
  <xsl:variable name="prefixL">
    <xsl:call-template name="hyphenPrefix">
      <xsl:with-param name="text" select="$subfield6l"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="displayL">
    <xsl:call-template name="getChron">
      <xsl:with-param name="caption" select="$subfield5l"/>
      <xsl:with-param name="text" select="$prefixL"/>
      <xsl:with-param name="lang" select="$lang"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$chronologyLevelSeparator"/>
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5l"/>
  </xsl:call-template>
  <xsl:value-of select="$displayL"/>
</xsl:if>

```

+++++++ Snippet End ++++++++

Action: Change lines 1088 to 1106 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:if test="string-length($subfield5k) and string-length($subfield6k)">
  <xsl:variable name="prefixK">
    <xsl:call-template name="hyphenPrefix">
      <xsl:with-param name="text" select="$subfield6k"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="displayK">
    <xsl:call-template name="getChron">
      <xsl:with-param name="caption" select="$subfield5k"/>
      <xsl:with-param name="text" select="$prefixK"/>
      <xsl:with-param name="lang" select="$lang"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$chronologyLevelSeparator"/>
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5k"/>
  </xsl:call-template>
  <xsl:value-of select="$displayK"/>
</xsl:if>

```

+++++++ Snippet End ++++++++

Action: Change lines 1068 to 1086 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:if test="string-length($subfield5j) and string-length($subfield6j)">

```

```

<xsl:variable name="prefixJ">
  <xsl:call-template name="hyphenPrefix">
    <xsl:with-param name="text" select="$subfield6j"/>
  </xsl:call-template>
</xsl:variable>
<xsl:variable name="displayJ">
  <xsl:call-template name="getChron">
    <xsl:with-param name="caption" select="$subfield5j"/>
    <xsl:with-param name="text" select="$prefixJ"/>
    <xsl:with-param name="lang" select="$lang"/>
  </xsl:call-template>
</xsl:variable>
<xsl:value-of select="$chronologyLevelSeparator"/>
<xsl:call-template name="displayChronCaption">
  <xsl:with-param name="caption" select="$subfield5j"/>
</xsl:call-template>
<xsl:value-of select="$displayJ"/>
</xsl:if>

```

+++++++ Snippet End ++++++

Action: Change lines 1049 to 1066 to the following snippet

+++++++ Snippet Start ++++++

```

<xsl:if test="string-length($subfield5i) and string-length($subfield6i)">
  <xsl:variable name="prefixI">
    <xsl:call-template name="hyphenPrefix">
      <xsl:with-param name="text" select="$subfield6i"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="displayI">
    <xsl:call-template name="getChron">
      <xsl:with-param name="caption" select="$subfield5i"/>
      <xsl:with-param name="text" select="$prefixI"/>
      <xsl:with-param name="lang" select="$lang"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5i"/>
  </xsl:call-template>
  <xsl:value-of select="$displayI"/>
</xsl:if>

```

+++++++ Snippet End ++++++

Action: Change lines 1039 to 1047 to the following snippet

+++++++ Snippet Start ++++++

```

<xsl:param name="lang"/>
<xsl:param name="subfield5i"/>
<xsl:param name="subfield5j"/>
<xsl:param name="subfield5k"/>
<xsl:param name="subfield5l"/>
<xsl:param name="subfield6i"/>
<xsl:param name="subfield6j"/>
<xsl:param name="subfield6k"/>
<xsl:param name="subfield6l"/>

```

+++++++ Snippet End ++++++

Action: Change lines 1024 to 1034 to the following snippet

+++++++ Snippet Start ++++++

```

<xsl:if test="string-length($suffix)">
  <xsl:if test="number($previous) > 0">
    <xsl:value-of select="$enumLevelSeparator"/>
  </xsl:if>

```

```

    <xsl:value-of select="$caption"/>
    <xsl:if test="substring($caption, (string-length($caption) - string-length($enumLabelDataSeparator)) + 1)
!= $enumLabelDataSeparator">
      <xsl:value-of select="$enumLabelDataSeparator"/>
    </xsl:if>
    <xsl:value-of select="$suffix"/>
  </xsl:if>
</xsl:if>
+++++++ Snippet End ++++++++

```

Action: Change lines 1016 to 1022 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($data)">
    <xsl:variable name="suffix">
      <xsl:call-template name="hyphenSuffixDedupe">
        <xsl:with-param name="text" select="$data"/>
        <xsl:with-param name="previous" select="$previous"/>
      </xsl:call-template>
    </xsl:variable>
  </xsl:if>
+++++++ Snippet End ++++++++

```

Action: Change lines 1012 to 1014 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:param name="caption"/>
  <xsl:param name="data"/>
  <xsl:param name="previous" select="0"/>
+++++++ Snippet End ++++++++

```

Action: Change lines 991 to 1005 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($subfield6f)">
    <xsl:variable name="displayF">
      <xsl:call-template name="concatEnum2">
        <xsl:with-param name="caption" select="$subfield5f"/>
        <xsl:with-param name="data" select="$subfield6f"/>
        <xsl:with-param name="previous" select="number(string-length($displayA) + string-
length($displayB) + string-length($displayC) + string-length($displayD) + string-length($displayE))"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:value-of select="$displayF"/>
  </xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>
+++++++ Snippet End ++++++++

```

Action: Change lines 981 to 989 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($subfield6e)">
    <xsl:variable name="displayE">
      <xsl:call-template name="concatEnum2">
        <xsl:with-param name="caption" select="$subfield5e"/>
        <xsl:with-param name="data" select="$subfield6e"/>
        <xsl:with-param name="previous" select="number(string-length($displayA) + string-
length($displayB) + string-length($displayC) + string-length($displayD))"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:value-of select="$displayE"/>
  </xsl:if>
+++++++ Snippet End ++++++++

```

Action: Change lines 971 to 979 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($subfield6d)">
  <xsl:variable name="displayD">
    <xsl:call-template name="concatEnum2">
      <xsl:with-param name="caption" select="$subfield5d"/>
      <xsl:with-param name="data" select="$subfield6d"/>
      <xsl:with-param name="previous" select="number(string-length($displayA) + string-
length($displayB) + string-length($displayC))"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$displayD"/>
+++++++ Snippet End +++++++
```

Action: Change lines 961 to 969 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($subfield6c)">
  <xsl:variable name="displayC">
    <xsl:call-template name="concatEnum2">
      <xsl:with-param name="caption" select="$subfield5c"/>
      <xsl:with-param name="data" select="$subfield6c"/>
      <xsl:with-param name="previous" select="number(string-length($displayA) + string-
length($displayB))"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$displayC"/>
+++++++ Snippet End +++++++
```

Action: Change lines 951 to 959 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($subfield6b)">
  <xsl:variable name="displayB">
    <xsl:call-template name="concatEnum2">
      <xsl:with-param name="caption" select="$subfield5b"/>
      <xsl:with-param name="data" select="$subfield6b"/>
      <xsl:with-param name="previous" select="string-length($displayA)"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$displayB"/>
+++++++ Snippet End +++++++
```

Action: Change lines 942 to 949 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($subfield6a)">
  <xsl:variable name="displayA">
    <xsl:call-template name="concatEnum2">
      <xsl:with-param name="caption" select="$subfield5a"/>
      <xsl:with-param name="data" select="$subfield6a"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:value-of select="$displayA"/>
+++++++ Snippet End +++++++
```

Action: Change lines 929 to 940 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:param name="subfield5a"/>
<xsl:param name="subfield5b"/>
<xsl:param name="subfield5c"/>
<xsl:param name="subfield5d"/>
<xsl:param name="subfield5e"/>
```

```
<xsl:param name="subfield5f"/>
<xsl:param name="subfield6a"/>
<xsl:param name="subfield6b"/>
<xsl:param name="subfield6c"/>
<xsl:param name="subfield6d"/>
<xsl:param name="subfield6e"/>
<xsl:param name="subfield6f"/>
+++++++ Snippet End +++++++
```

Action: Change lines 904 to 923 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:param name="caption"/>
<xsl:param name="data"/>
<xsl:param name="previous" select="true()"/>
<xsl:if test="string-length($data)">
  <xsl:variable name="prefix">
    <xsl:call-template name="hyphenPrefix">
      <xsl:with-param name="text" select="$data"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:if test="string-length($prefix)">
    <xsl:if test="$previous = true()">
      <xsl:value-of select="$enumLevelSeparator"/>
    </xsl:if>
    <xsl:value-of select="$caption"/>
    <xsl:if test="substring($caption, (string-length($caption) - string-length($enumLabelDataSeparator)) + 1)
!= $enumLabelDataSeparator">
      <xsl:value-of select="$enumLabelDataSeparator"/>
    </xsl:if>
    <xsl:value-of select="$prefix"/>
  </xsl:if>
</xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 894 to 897 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:call-template name="concatEnum1">
  <xsl:with-param name="caption" select="$subfield5f"/>
  <xsl:with-param name="data" select="$subfield6f"/>
</xsl:call-template>
+++++++ Snippet End ++++++++
```

Action: Change lines 889 to 892 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:call-template name="concatEnum1">
  <xsl:with-param name="caption" select="$subfield5e"/>
  <xsl:with-param name="data" select="$subfield6e"/>
</xsl:call-template>
+++++++ Snippet End ++++++++
```

Action: Change lines 884 to 887 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:call-template name="concatEnum1">
  <xsl:with-param name="caption" select="$subfield5d"/>
  <xsl:with-param name="data" select="$subfield6d"/>
</xsl:call-template>
+++++++ Snippet End ++++++++
```

Action: Change lines 879 to 882 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:call-template name="concatEnum1">
```



```
    <xsl:with-param name="caption" select="$subfield5c"/>
    <xsl:with-param name="data" select="$subfield6c"/>
  </xsl:call-template>
+++++++ Snippet End +++++++
```

Action: Change lines 874 to 877 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:call-template name="concatEnum1">
    <xsl:with-param name="caption" select="$subfield5b"/>
    <xsl:with-param name="data" select="$subfield6b"/>
  </xsl:call-template>
+++++++ Snippet End ++++++++
```

Action: Change lines 868 to 872 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:call-template name="concatEnum1">
    <xsl:with-param name="caption" select="$subfield5a"/>
    <xsl:with-param name="data" select="$subfield6a"/>
    <xsl:with-param name="previous" select="false()"/>
  </xsl:call-template>
+++++++ Snippet End ++++++++
```

Action: Change lines 855 to 866 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:param name="subfield5a"/>
  <xsl:param name="subfield5b"/>
  <xsl:param name="subfield5c"/>
  <xsl:param name="subfield5d"/>
  <xsl:param name="subfield5e"/>
  <xsl:param name="subfield5f"/>
  <xsl:param name="subfield6a"/>
  <xsl:param name="subfield6b"/>
  <xsl:param name="subfield6c"/>
  <xsl:param name="subfield6d"/>
  <xsl:param name="subfield6e"/>
  <xsl:param name="subfield6f"/>
+++++++ Snippet End ++++++++
```

Action: Change lines 836 to 848 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:if test="$whichPair != '3'">
    <xsl:choose>
      <xsl:when test="$whichPair = '4'">
        <xsl:value-of select="$supplementCaption"/>
      </xsl:when>
      <xsl:when test="$whichPair = '5'">
        <xsl:value-of select="$indexCaption"/>
      </xsl:when>
    </xsl:choose>
    <xsl:if test="string-length($subfield5o)">
      <xsl:value-of select="concat(' - ', $subfield5o)"/>
    </xsl:if>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 833 to 834 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:param name="whichPair"/>
  <xsl:param name="subfield5o"/>
+++++++ Snippet End ++++++++
```

Action: Add the following snippet at starting at line 827

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($suffix)">
  <xsl:value-of select="$chronologyRangeSeparator"/>
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5m"/>
  </xsl:call-template>
  <xsl:call-template name="getChron">
    <xsl:with-param name="caption" select="$subfield5m"/>
    <xsl:with-param name="text" select="$suffix"/>
    <xsl:with-param name="lang" select="$lang"/>
  </xsl:call-template>
</xsl:if>
```

</xsl:if>

+++++++ Snippet End +++++++

Action: Change line 826 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($prefix)">
  <xsl:call-template name="displayChronCaption">
    <xsl:with-param name="caption" select="$subfield5m"/>
  </xsl:call-template>
  <xsl:call-template name="getChron">
    <xsl:with-param name="caption" select="$subfield5m"/>
    <xsl:with-param name="text" select="$prefix"/>
    <xsl:with-param name="lang" select="$lang"/>
  </xsl:call-template>
</xsl:if>
```

+++++++ Snippet End +++++++

Action: Change lines 814 to 824 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="suffix">
  <xsl:call-template name="hyphenSuffixDedupe">
    <xsl:with-param name="text" select="$subfield6m"/>
  </xsl:call-template>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 803 to 812 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="prefix">
  <xsl:call-template name="hyphenPrefix">
    <xsl:with-param name="text" select="$subfield6m"/>
  </xsl:call-template>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 789 to 801 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:if test="string-length($subfield5m) and string-length($subfield6m)">
```

+++++++ Snippet End +++++++

Action: Change lines 785 to 787 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:param name="subfield5m"/>
<xsl:param name="subfield6m"/>
<xsl:param name="lang"/>
```

+++++++ Snippet End +++++++

```
#### Action: Add the following snippet at starting at line 779 ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($subfield5h) and string-length($subfield6h)">
    <xsl:value-of select="$subfield5h"/>
    <xsl:if test="substring($subfield5h, (string-length($subfield5h) - string-length($enumLabelDataSeparator)) +
1) != $enumLabelDataSeparator">
      <xsl:value-of select="$enumLabelDataSeparator"/>
    </xsl:if>
    <xsl:value-of select="$subfield6h"/>
  </xsl:if>
```

+++++++ Snippet End ++++++++

```
#### Action: Change lines 772 to 778 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($subfield5g) and string-length($subfield6g) and string-length($subfield5h) and
string-length($subfield6h)">
    <xsl:value-of select="$enumLevelSeparator"/>
  </xsl:if>
```

+++++++ Snippet End ++++++++

```
#### Action: Change lines 768 to 770 to the following snippet ####
+++++++ Snippet Start ++++++++
  <xsl:if test="string-length($subfield5g) and string-length($subfield6g)">
    <xsl:value-of select="$subfield5g"/>
    <xsl:if test="substring($subfield5g, (string-length($subfield5g) - string-length($enumLabelDataSeparator)) +
1) != $enumLabelDataSeparator">
      <xsl:value-of select="$enumLabelDataSeparator"/>
    </xsl:if>
    <xsl:value-of select="$subfield6g"/>
  </xsl:if>
```

+++++++ Snippet End ++++++++

```
#### Action: Change lines 760 to 766 to the following snippet ####
+++++++ Snippet Start ++++++++
  <!-- there is no collapsing of alternative enumerations -->
```

+++++++ Snippet End ++++++++

```
#### Action: Change line 758 to the following snippet ####
```

```
+++++++ Snippet Start ++++++++
  <xsl:param name="subfield5g"/>
  <xsl:param name="subfield5h"/>
  <xsl:param name="subfield6g"/>
  <xsl:param name="subfield6h"/>
```

+++++++ Snippet End ++++++++

```
#### Action: Delete lines 752 - 756 ####
```

```
#### Action: Add the following snippet at starting at line 746 ####
```

```
+++++++ Snippet Start ++++++++
  <xsl:variable name="caption">
    <xsl:call-template name="buildCaptionVariable">
      <xsl:with-param name="whichPair" select="$whichPair"/>
      <xsl:with-param name="subfield5o" select="$subfield5o"/>
    </xsl:call-template>
  </xsl:variable>
```

<!-- Build the Basic Unit (85*/86*) display -->

```
<xsl:choose>
```

```

<xsl:when test="translate($displayFormat, $lowercase, $uppercase)='STANDARD'">
  <xsl:call-template name="standardFormat">
    <xsl:with-param name="enum1" select="$enum1"/>
    <xsl:with-param name="enum2" select="$enum2"/>
    <xsl:with-param name="chron1" select="$chron1"/>
    <xsl:with-param name="chron2" select="$chron2"/>
    <xsl:with-param name="altEnum" select="$altEnum"/>
    <xsl:with-param name="altChron" select="$altChron"/>
    <xsl:with-param name="lineCaption" select="$caption"/>
  </xsl:call-template>
</xsl:when>
<xsl:when test="translate($displayFormat, $lowercase, $uppercase)='NISO'">
  <xsl:call-template name="NISOFormat">
    <xsl:with-param name="enum1" select="$enum1"/>
    <xsl:with-param name="enum2" select="$enum2"/>
    <xsl:with-param name="chron1" select="$chron1"/>
    <xsl:with-param name="chron2" select="$chron2"/>
    <xsl:with-param name="altEnum" select="$altEnum"/>
    <xsl:with-param name="altChron" select="$altChron"/>
    <xsl:with-param name="lineCaption" select="$caption"/>
  </xsl:call-template>
</xsl:when>
</xsl:choose>

```

+++++++ Snippet End +++++++

Action: Change lines 720 to 745 to the following snippet

+++++++ Snippet Start +++++++

```

<xsl:variable name="altChron">
  <xsl:choose>
    <xsl:when test="$areChronValuesInEnumerativeSubfields='false'">
      <xsl:call-template name="buildAltChronVariable">
        <xsl:with-param name="subfield5m" select="$subfield5m"/>
        <xsl:with-param name="subfield6m" select="$subfield6m"/>
        <xsl:with-param name="lang" select="$lang"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="buildAltChronVariable">
        <xsl:with-param name="subfield5m" select="$subfield5e"/>
        <xsl:with-param name="subfield6m" select="$subfield6e"/>
        <xsl:with-param name="lang" select="$lang"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

```

+++++++ Snippet End +++++++

Action: Change lines 694 to 718 to the following snippet

+++++++ Snippet Start +++++++

```

<xsl:variable name="altEnum">
  <xsl:choose>
    <xsl:when test="$areChronValuesInEnumerativeSubfields='false'">
      <xsl:call-template name="buildAltEnumVariable">
        <xsl:with-param name="subfield5g" select="$subfield5g"/>
        <xsl:with-param name="subfield5h" select="$subfield5h"/>
        <xsl:with-param name="subfield6g" select="$subfield6g"/>
        <xsl:with-param name="subfield6h" select="$subfield6h"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>

```

```

        <xsl:value-of select="""/>
    </xsl:otherwise>
</xsl:choose>
</xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 678 to 692 to the following snippet

```

+++++++ Snippet Start ++++++
    <xsl:variable name="chron2">
        <xsl:choose>
            <xsl:when test="$areChronValuesInEnumerativeSubfields='false'">
                <xsl:call-template name="buildChron2Variable">
                    <xsl:with-param name="lang" select="$lang"/>
                    <xsl:with-param name="subfield5i" select="$subfield5i"/>
                    <xsl:with-param name="subfield5j" select="$subfield5j"/>
                    <xsl:with-param name="subfield5k" select="$subfield5k"/>
                    <xsl:with-param name="subfield5l" select="$subfield5l"/>
                    <xsl:with-param name="subfield6i" select="$subfield6i"/>
                    <xsl:with-param name="subfield6j" select="$subfield6j"/>
                    <xsl:with-param name="subfield6k" select="$subfield6k"/>
                    <xsl:with-param name="subfield6l" select="$subfield6l"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:call-template name="buildChron2Variable">
                    <xsl:with-param name="lang" select="$lang"/>
                    <xsl:with-param name="subfield5i" select="$subfield5a"/>
                    <xsl:with-param name="subfield5j" select="$subfield5b"/>
                    <xsl:with-param name="subfield5k" select="$subfield5c"/>
                    <xsl:with-param name="subfield5l" select="$subfield5d"/>
                    <xsl:with-param name="subfield6i" select="$subfield6a"/>
                    <xsl:with-param name="subfield6j" select="$subfield6b"/>
                    <xsl:with-param name="subfield6k" select="$subfield6c"/>
                    <xsl:with-param name="subfield6l" select="$subfield6d"/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 647 to 676 to the following snippet

```

+++++++ Snippet Start ++++++
    <xsl:variable name="chron1">
        <xsl:choose>
            <xsl:when test="$areChronValuesInEnumerativeSubfields='false'">
                <xsl:call-template name="buildChron1Variable">
                    <xsl:with-param name="lang" select="$lang"/>
                    <xsl:with-param name="subfield5i" select="$subfield5i"/>
                    <xsl:with-param name="subfield5j" select="$subfield5j"/>
                    <xsl:with-param name="subfield5k" select="$subfield5k"/>
                    <xsl:with-param name="subfield5l" select="$subfield5l"/>
                    <xsl:with-param name="subfield6i" select="$subfield6i"/>
                    <xsl:with-param name="subfield6j" select="$subfield6j"/>
                    <xsl:with-param name="subfield6k" select="$subfield6k"/>
                    <xsl:with-param name="subfield6l" select="$subfield6l"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:call-template name="buildChron1Variable">
                    <xsl:with-param name="lang" select="$lang"/>
                    <xsl:with-param name="subfield5i" select="$subfield5a"/>

```

```

    <xsl:with-param name="subfield5j" select="$subfield5b"/>
    <xsl:with-param name="subfield5k" select="$subfield5c"/>
    <xsl:with-param name="subfield5l" select="$subfield5d"/>
    <xsl:with-param name="subfield6i" select="$subfield6a"/>
    <xsl:with-param name="subfield6j" select="$subfield6b"/>
    <xsl:with-param name="subfield6k" select="$subfield6c"/>
    <xsl:with-param name="subfield6l" select="$subfield6d"/>
  </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
+++++++ Snippet End ++++++++

```

Action: Change lines 616 to 645 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:variable name="enum2">
  <xsl:choose>
    <xsl:when test="$areChronValuesInEnumerativeSubfields='false'">
      <xsl:call-template name="buildEnum2Variable">
        <xsl:with-param name="subfield5a" select="$subfield5a"/>
        <xsl:with-param name="subfield5b" select="$subfield5b"/>
        <xsl:with-param name="subfield5c" select="$subfield5c"/>
        <xsl:with-param name="subfield5d" select="$subfield5d"/>
        <xsl:with-param name="subfield5e" select="$subfield5e"/>
        <xsl:with-param name="subfield5f" select="$subfield5f"/>
        <xsl:with-param name="subfield6a" select="$subfield6a"/>
        <xsl:with-param name="subfield6b" select="$subfield6b"/>
        <xsl:with-param name="subfield6c" select="$subfield6c"/>
        <xsl:with-param name="subfield6d" select="$subfield6d"/>
        <xsl:with-param name="subfield6e" select="$subfield6e"/>
        <xsl:with-param name="subfield6f" select="$subfield6f"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select=""/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
+++++++ Snippet End ++++++++

```

Action: Change lines 568 to 614 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:variable name="enum1">
  <xsl:choose>
    <xsl:when test="$areChronValuesInEnumerativeSubfields='false'">
      <xsl:call-template name="buildEnum1Variable">
        <xsl:with-param name="subfield5a" select="$subfield5a"/>
        <xsl:with-param name="subfield5b" select="$subfield5b"/>
        <xsl:with-param name="subfield5c" select="$subfield5c"/>
        <xsl:with-param name="subfield5d" select="$subfield5d"/>
        <xsl:with-param name="subfield5e" select="$subfield5e"/>
        <xsl:with-param name="subfield5f" select="$subfield5f"/>
        <xsl:with-param name="subfield6a" select="$subfield6a"/>
        <xsl:with-param name="subfield6b" select="$subfield6b"/>
        <xsl:with-param name="subfield6c" select="$subfield6c"/>
        <xsl:with-param name="subfield6d" select="$subfield6d"/>
        <xsl:with-param name="subfield6e" select="$subfield6e"/>
        <xsl:with-param name="subfield6f" select="$subfield6f"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>

```

```

        <xsl:value-of select="""/>
    </xsl:otherwise>
</xsl:choose>
</xsl:variable>
+++++++ Snippet End ++++++++

```

Action: Change lines 536 to 566 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:param name="areChronValuesInEnumerativeSubfields"/>
<xsl:param name="lang"/>
<xsl:param name="whichPair"/>      <!-- send '3' or '4' or '5' -->
<xsl:param name="subfield5a"/>    <!-- s85*a -->
<xsl:param name="subfield5b"/>    <!-- s85*b -->
<xsl:param name="subfield5c"/>    <!-- s85*c -->
<xsl:param name="subfield5d"/>    <!-- s85*d -->
<xsl:param name="subfield5e"/>    <!-- s85*e -->
<xsl:param name="subfield5f"/>    <!-- s85*f -->
<xsl:param name="subfield5g"/>    <!-- s85*g -->
<xsl:param name="subfield5h"/>    <!-- s85*h -->
<xsl:param name="subfield5i"/>    <!-- s85*i -->
<xsl:param name="subfield5j"/>    <!-- s85*j -->
<xsl:param name="subfield5k"/>    <!-- s85*k -->
<xsl:param name="subfield5l"/>    <!-- s85*l -->
<xsl:param name="subfield5m"/>    <!-- s85*m -->
<xsl:param name="subfield5o"/>    <!-- s85*o -->
<xsl:param name="subfield6a"/>    <!-- s86*a -->
<xsl:param name="subfield6b"/>    <!-- s86*b -->
<xsl:param name="subfield6c"/>    <!-- s86*c -->
<xsl:param name="subfield6d"/>    <!-- s86*d -->
<xsl:param name="subfield6e"/>    <!-- s86*e -->
<xsl:param name="subfield6f"/>    <!-- s86*f -->
<xsl:param name="subfield6g"/>    <!-- s86*g -->
<xsl:param name="subfield6h"/>    <!-- s86*h -->
<xsl:param name="subfield6i"/>    <!-- s86*i -->
<xsl:param name="subfield6j"/>    <!-- s86*j -->
<xsl:param name="subfield6k"/>    <!-- s86*k -->
<xsl:param name="subfield6l"/>    <!-- s86*l -->
<xsl:param name="subfield6m"/>    <!-- s86*m -->
<xsl:param name="subfield6o"/>    <!-- s86*o -->
+++++++ Snippet End ++++++++

```

Action: Add the following snippet at starting at line 532

```

+++++++ Snippet Start ++++++++
<xsl:variable name="subfield5clsEnum">
  <xsl:choose>
    <xsl:when test="string-length($subfield5c)=0">
      <xsl:value-of select=""false""/>
    </xsl:when>
    <xsl:when test="(starts-with($subfield5c, '(') and (substring($subfield5c, string-length($subfield5c)) =
'))">
      <xsl:value-of select=""false""/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select=""true""/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="subfield5dlsEnum">
  <xsl:choose>
    <xsl:when test="string-length($subfield5d)=0">

```

```

        <xsl:value-of select=""false""/>
    </xsl:when>
    <xsl:when test="(starts-with($subfield5d, '(') and (substring($subfield5d, string-length($subfield5d)) =
'))">
        <xsl:value-of select=""false""/>
    </xsl:when>
    <xsl:otherwise>
        <xsl:value-of select=""true""/>
    </xsl:otherwise>
    </xsl:choose>
</xsl:variable>

<xsl:variable name="subfield5elsEnum">
    <xsl:choose>
        <xsl:when test="string-length($subfield5e)=0">
            <xsl:value-of select=""false""/>
        </xsl:when>
        <xsl:when test="(starts-with($subfield5e, '(') and (substring($subfield5e, string-length($subfield5e)) =
'))">
            <xsl:value-of select=""false""/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select=""true""/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:variable>

<xsl:variable name="subfield5flsEnum">
    <xsl:choose>
        <xsl:when test="string-length($subfield5f)=0">
            <xsl:value-of select=""false""/>
        </xsl:when>
        <xsl:when test="(starts-with($subfield5f, '(') and (substring($subfield5f, string-length($subfield5f)) = '))">
            <xsl:value-of select=""false""/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select=""true""/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:variable>

    <xsl:variable name="lengthOfSubfieldIJKL" select="string-length(concat($subfield5i, $subfield5j, $subfield5k,
$subfield5l, $subfield5m))"/>

    <xsl:choose>
        <xsl:when test="contains(concat($subfield5alsEnum, $subfield5blsEnum,
$subfield5clsEnum,$subfield5dlsEnum,$subfield5elsEnum, $subfield5flsEnum), 'true')">
            <xsl:value-of select = ""false""/>
        </xsl:when>
        <xsl:when test="$lengthOfSubfieldIJKL!=0">
            <xsl:value-of select = ""false""/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select = ""true""/>
        </xsl:otherwise>
    </xsl:choose>

</xsl:template>

+++++++ Snippet End ++++++++

```


Action: Change lines 517 to 531 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="subfield5blsEnum">
  <xsl:choose>
    <xsl:when test="string-length($subfield5b)=0">
      <xsl:value-of select="false"/>
    </xsl:when>
    <xsl:when test="(starts-with($subfield5b, '(') and (substring($subfield5b, string-length($subfield5b)) =
'))">
      <xsl:value-of select="false"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="true"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 433 to 515 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="subfield5alsEnum">
  <xsl:choose>
    <xsl:when test="string-length($subfield5a)=0">
      <xsl:value-of select="false"/>
    </xsl:when>
    <xsl:when test="(starts-with($subfield5a, '(') and (substring($subfield5a, string-length($subfield5a)) =
'))">
      <xsl:value-of select="false"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="true"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 421 to 431 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:param name="subfield5a"/>
<xsl:param name="subfield5b"/>
<xsl:param name="subfield5c"/>
<xsl:param name="subfield5d"/>
<xsl:param name="subfield5e"/>
<xsl:param name="subfield5f"/>
<xsl:param name="subfield5i"/>
<xsl:param name="subfield5j"/>
<xsl:param name="subfield5k"/>
<xsl:param name="subfield5l"/>
<xsl:param name="subfield5m"/>
```

+++++++ Snippet End +++++++

Action: Change lines 410 to 416 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:param name="subfieldA"/>
<xsl:param name="subfieldZ"/>
<xsl:value-of select="$subfieldA"/>
<xsl:if test="string-length($subfieldA)>0 and string-length($subfieldZ)>0">
  <xsl:text> </xsl:text>
</xsl:if>
<xsl:value-of select="$subfieldZ"/>
```

+++++++ Snippet End +++++++

Action: Change lines 397 to 404 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:choose>
  <xsl:when test="contains($ind1MustBe, $ind1) and contains($ind2MustBe, $ind2)">
    <xsl:value-of select="true"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select="false"/>
  </xsl:otherwise>
</xsl:choose>
```

+++++++ Snippet End +++++++

Action: Change lines 383 to 395 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="ind2MustBe">
  <xsl:choose>
    <xsl:when test="$whichPair='3'">
      <xsl:value-of select="$Valid863Indicators_2"/>
    </xsl:when>
    <xsl:when test="$whichPair='4'">
      <xsl:value-of select="$Valid864Indicators_2"/>
    </xsl:when>
    <xsl:when test="$whichPair='5'">
      <xsl:value-of select="$Valid865Indicators_2"/>
    </xsl:when>
  </xsl:choose>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 369 to 381 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="ind1MustBe">
  <xsl:choose>
    <xsl:when test="$whichPair='3'">
      <xsl:value-of select="$Valid863Indicators_1"/>
    </xsl:when>
    <xsl:when test="$whichPair='4'">
      <xsl:value-of select="$Valid864Indicators_1"/>
    </xsl:when>
    <xsl:when test="$whichPair='5'">
      <xsl:value-of select="$Valid865Indicators_1"/>
    </xsl:when>
  </xsl:choose>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change lines 362 to 367 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:variable name="ind1">
  <xsl:value-of select="@ind1"/>
</xsl:variable>
<xsl:variable name="ind2">
  <xsl:value-of select="@ind2"/>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Change line 360 to the following snippet

+++++++ Snippet Start +++++++

```
<xsl:param name="whichPair"/>
```

+++++++ Snippet End +++++++

Action: Change lines 301 to 354 to the following snippet

+++++++ Snippet Start +++++++

<!-- Display preferred [866|867|868] data fields if they exist -->

```
<xsl:if test="(@tag)=$preferredDataFieldsTag">
  <xsl:variable name="preferredHoldings">
    <xsl:call-template name="buildPreferredDataFields">
      <xsl:with-param name="subfieldA" select="slim:subfield[@code='a']"/>
      <xsl:with-param name="subfieldZ" select="slim:subfield[@code='z']"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:if test="string-length($preferredHoldings)>0">
    <xsl:value-of select="$preferredHoldings"/><br/>
  </xsl:if>
</xsl:if>
```

<!-- 1st test to display 85X/86X pairs: The linking numbers must match. -->

```
<xsl:if test="(@tag=$optionalDataFieldsTag) and (floor(slim:subfield[@code='8']) = $linking_number)">
```

<!-- 2nd test to display 85X/86X pairs: None of the 866|867|868 fields may have a link value of zero. -->

```
<xsl:if test="$countPreferredDataFieldsWithMatchingLink=0">
```

```
  <xsl:variable name="indicatorTest">
    <xsl:call-template name="checkIndicators">
      <xsl:with-param name="whichPair" select="$whichPair"/>
    </xsl:call-template>
  </xsl:variable>
```

```
  <xsl:if test="$indicatorTest='true'">
    <xsl:variable name="optionalHoldings">
      <xsl:call-template name="buildHoldingsVariableForOptionalDataFields">
        <xsl:with-param name="areChronValuesInEnumerativeSubfields"
select="$areChronValuesInEnumerativeSubfields"/>
```

```
      <xsl:with-param name="lang" select="$languageCode"/>
      <xsl:with-param name="whichPair" select="$whichPair"/>
      <xsl:with-param name="subfield5a" select="$captionA"/>
      <xsl:with-param name="subfield5b" select="$captionB"/>
      <xsl:with-param name="subfield5c" select="$captionC"/>
      <xsl:with-param name="subfield5d" select="$captionD"/>
      <xsl:with-param name="subfield5e" select="$captionE"/>
      <xsl:with-param name="subfield5f" select="$captionF"/>
      <xsl:with-param name="subfield5g" select="$captionG"/>
      <xsl:with-param name="subfield5h" select="$captionH"/>
      <xsl:with-param name="subfield5i" select="$captionI"/>
      <xsl:with-param name="subfield5j" select="$captionJ"/>
      <xsl:with-param name="subfield5k" select="$captionK"/>
      <xsl:with-param name="subfield5l" select="$captionL"/>
      <xsl:with-param name="subfield5m" select="$captionM"/>
      <xsl:with-param name="subfield5o" select="$captionO"/>
      <xsl:with-param name="subfield6a" select="slim:subfield[@code='a']"/>
      <xsl:with-param name="subfield6b" select="slim:subfield[@code='b']"/>
      <xsl:with-param name="subfield6c" select="slim:subfield[@code='c']"/>
      <xsl:with-param name="subfield6d" select="slim:subfield[@code='d']"/>
      <xsl:with-param name="subfield6e" select="slim:subfield[@code='e']"/>
      <xsl:with-param name="subfield6f" select="slim:subfield[@code='f']"/>
      <xsl:with-param name="subfield6g" select="slim:subfield[@code='g']"/>
      <xsl:with-param name="subfield6h" select="slim:subfield[@code='h']"/>
      <xsl:with-param name="subfield6i" select="slim:subfield[@code='i']"/>
      <xsl:with-param name="subfield6j" select="slim:subfield[@code='j']"/>
```

```

        <xsl:with-param name="subfield6k" select="slim:subfield[@code='k']"/>
        <xsl:with-param name="subfield6l" select="slim:subfield[@code='l']"/>
        <xsl:with-param name="subfield6m" select="slim:subfield[@code='m']"/>
        <xsl:with-param name="subfield6o" select="slim:subfield[@code='o']"/>
    </xsl:call-template>
</xsl:variable>
<xsl:if test="string-length($optionalHoldings)>0">
    <xsl:value-of select="$optionalHoldings"/><br/>
</xsl:if>
</xsl:if> <!-- <xsl:if test="$indicatorTest='true'" --> -->
</xsl:if>
</xsl:if>
</xsl:for-each>
</xsl:for-each>
+++++++ Snippet End ++++++

```

Action: Change lines 297 to 299 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- For each matching data field -->
<xsl:for-each select="../slim:datafield[(@tag=$optionalDataFieldsTag or @tag=$preferredDataFieldsTag)]">
    <!-- Sort by sequence numbers -->
    <xsl:sort
        select="floor(slim:subfield[@code='8'])"
        order="ascending"
        data-type="number"/>
    <xsl:sort
        select="substring-after(slim:subfield[@code='8'], '.')"
        order="ascending"
        data-type="number"/>
+++++++ Snippet End ++++++

```

Action: Change lines 269 to 295 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- Get number of instances where [866|867|888] $8<linking number> == 85X $8<linking number> -->
<xsl:variable name="countPreferredDataFieldsWithMatchingLink"
select="count(slim:datafield[@tag=$preferredDataFieldsTag and
floor(slim:subfield[@code='8'])=$linking_number]"/>
+++++++ Snippet End ++++++

```

Action: Change lines 252 to 267 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- Determine if rogue cataloger put chronological fields in enumerative fields. -->
<xsl:variable name="areChronValuesInEnumerativeSubfields">
    <xsl:call-template name="chronValuesInEnumerativeSubfields">
        <xsl:with-param name="subfield5a" select="$captionA"/>
        <xsl:with-param name="subfield5b" select="$captionB"/>
        <xsl:with-param name="subfield5c" select="$captionC"/>
        <xsl:with-param name="subfield5d" select="$captionD"/>
        <xsl:with-param name="subfield5e" select="$captionE"/>
        <xsl:with-param name="subfield5f" select="$captionF"/>
        <xsl:with-param name="subfield5i" select="$captionI"/>
        <xsl:with-param name="subfield5j" select="$captionJ"/>
        <xsl:with-param name="subfield5k" select="$captionK"/>
        <xsl:with-param name="subfield5l" select="$captionL"/>
        <xsl:with-param name="subfield5m" select="$captionM"/>
    </xsl:call-template>
</xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 236 to 250 to the following snippet

```

+++++++ Snippet Start ++++++

```

```

<!-- get all the captions -->
<xsl:variable name="captionA"><xsl:value-of select="slim:subfield[@code='a']"/></xsl:variable>
<xsl:variable name="captionB"><xsl:value-of select="slim:subfield[@code='b']"/></xsl:variable>
<xsl:variable name="captionC"><xsl:value-of select="slim:subfield[@code='c']"/></xsl:variable>
<xsl:variable name="captionD"><xsl:value-of select="slim:subfield[@code='d']"/></xsl:variable>
<xsl:variable name="captionE"><xsl:value-of select="slim:subfield[@code='e']"/></xsl:variable>
<xsl:variable name="captionF"><xsl:value-of select="slim:subfield[@code='f']"/></xsl:variable>
<xsl:variable name="captionG"><xsl:value-of select="slim:subfield[@code='g']"/></xsl:variable>
<xsl:variable name="captionH"><xsl:value-of select="slim:subfield[@code='h']"/></xsl:variable>
<xsl:variable name="captionI"><xsl:value-of select="slim:subfield[@code='i']"/></xsl:variable>
<xsl:variable name="captionJ"><xsl:value-of select="slim:subfield[@code='j']"/></xsl:variable>
<xsl:variable name="captionK"><xsl:value-of select="slim:subfield[@code='k']"/></xsl:variable>
<xsl:variable name="captionL"><xsl:value-of select="slim:subfield[@code='l']"/></xsl:variable>
<xsl:variable name="captionM"><xsl:value-of select="slim:subfield[@code='m']"/></xsl:variable>
<xsl:variable name="captionO"><xsl:value-of select="slim:subfield[@code='o']"/></xsl:variable>
+++++++ Snippet End ++++++++

```

Action: Change lines 231 to 234 to the following snippet

```

+++++++ Snippet Start ++++++++
<!-- Get linking number from 85X $8 subfield -->
<xsl:variable name="linking_number">
  <xsl:value-of select="slim:subfield[@code='8']"/>
</xsl:variable>
+++++++ Snippet End ++++++++

```

Action: Change lines 225 to 229 to the following snippet

```

+++++++ Snippet Start ++++++++
<!-- sort by 85X [853|854|855] linking numbers -->
<xsl:sort
  select="slim:subfield[@code='8']"
  order="ascending"
  data-type="number"/>
+++++++ Snippet End ++++++++

```

Action: Change lines 222 to 223 to the following snippet

```

+++++++ Snippet Start ++++++++
<!-- For each matching caption -->
<xsl:for-each select="slim:datafield[@tag=$captionFieldsTag]">
+++++++ Snippet End ++++++++

```

Action: Change lines 215 to 220 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:param name="languageCode"/>
<xsl:param name="captionFieldsTag"/>
<xsl:param name="optionalDataFieldsTag"/>
<xsl:param name="preferredDataFieldsTag"/>
<xsl:param name="countPreferredDataFieldsWithLinkValues0"/>
<xsl:param name="whichPair"/>
+++++++ Snippet End ++++++++

```

Action: Change lines 206 to 210 to the following snippet

```

+++++++ Snippet Start ++++++++
<xsl:param name="preferredDataFieldsTag"/>
<xsl:call-template name="buildPreferredDataFields">
  <xsl:with-param name="subfieldA" select="slim:subfield[@code='a']"/>
  <xsl:with-param name="subfieldZ" select="slim:subfield[@code='z']"/>
</xsl:call-template>
+++++++ Snippet End ++++++++

```

Action: Change lines 194 to 201 to the following snippet

```

+++++++ Snippet Start ++++++++

```

```

<xsl:param name="preferredDataFieldsTag"/>
<xsl:for-each select="slim:datafield[@tag=$preferredDataFieldsTag and slim:subfield='0' and
slim:subfield[@code='8']]">
  <xsl:call-template name="buildPreferredDataFields">
    <xsl:with-param name="subfieldA" select="slim:subfield[@code='a']"/>
    <xsl:with-param name="subfieldZ" select="slim:subfield[@code='z']"/>
  </xsl:call-template>
  <br/>
</xsl:for-each>
+++++++ Snippet End ++++++

```

Action: Change line 189 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- Parse and display [866|867|868] fields that have a linking value of zero -->
<xsl:if test="$countPreferredDataFieldsWithLinkValues0 != 0">
  <xsl:call-template name="parseAndDisplayAllPreferredDataFields">
    <xsl:with-param name="preferredDataFieldsTag" select="$preferredDataFieldsTag"/>
  </xsl:call-template>
</xsl:if>

<!-- Parse and display all linked pairs of 85X/86X fields, and unlinked [866|867|868] fields, but only if
NONE of the [866|867|868] have linking values of zero -->
<xsl:if test="$countPreferredDataFieldsWithLinkValues0 = 0">
  <xsl:call-template name="parseAndDisplay85Xand86XPairs">
    <xsl:with-param name="languageCode" select="$languageCode"/>
    <xsl:with-param name="captionFieldsTag" select="$captionFieldsTag"/>
    <xsl:with-param name="optionalDataFieldsTag" select="$optionalDataFieldsTag"/>
    <xsl:with-param name="preferredDataFieldsTag" select="$preferredDataFieldsTag"/>
    <xsl:with-param name="countPreferredDataFieldsWithLinkValues0"
select="$countPreferredDataFieldsWithLinkValues0"/>
    <xsl:with-param name="whichPair" select="$whichPair"/>
  </xsl:call-template>
</xsl:if>

</xsl:if>
+++++++ Snippet End ++++++

```

Action: Change lines 176 to 187 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- Get count of [866|867|868] nodes that have a linking value of zero -->
<xsl:variable name="countPreferredDataFieldsWithLinkValues0">
  <xsl:value-of select="count(slim:datafield[@tag=$preferredDataFieldsTag and slim:subfield='0' and
slim:subfield[@code='8']])"/>
</xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 169 to 174 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- Select [866|867|868] tag identifier that over-rides [863|864|865] in some instances -->
<xsl:variable name="preferredDataFieldsTag">
  <xsl:value-of select="concat('86', string((number($whichPair)+3)))/>
</xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 164 to 167 to the following snippet

```

+++++++ Snippet Start ++++++
<!-- Select [863|864|865] tag identifier -->
<xsl:variable name="optionalDataFieldsTag">
  <xsl:value-of select="concat('86', $whichPair)/>
</xsl:variable>
+++++++ Snippet End ++++++

```

Action: Change lines 154 to 162 to the following snippet

```
+++++++ Snippet Start +++++++
  <!-- Select [853|854|855] tag identifier -->
  <xsl:variable name="captionFieldsTag">
    <xsl:value-of select="concat('85', $whichPair)"/>
  </xsl:variable>
+++++++ Snippet End +++++++
```

Action: Change lines 149 to 152 to the following snippet

```
+++++++ Snippet Start +++++++
  <!-- get and save language code from 008 -->
  <xsl:variable name="languageCode" select="substring(slim:controlfield[@tag='008'], 23, 3)"/>
+++++++ Snippet End +++++++
```

Action: Change lines 143 to 147 to the following snippet

```
+++++++ Snippet Start +++++++
  <!-- Validate whichPair value -->
  <xsl:if test="string-length($whichPair) = 1 and contains('345', $whichPair)">
+++++++ Snippet End +++++++
```

Action: Change line 141 to the following snippet

```
+++++++ Snippet Start +++++++
  <xsl:param name="whichPair"/>  <!-- send 3 or 4 or 5 -->
+++++++ Snippet End +++++++
```

Action: Change lines 133 to 135 to the following snippet

```
+++++++ Snippet Start +++++++
  <xsl:call-template name="parseAndDisplayConciseHoldings">
    <xsl:with-param name="whichPair" select="'5'"/>  <!-- display 855 / 865 pairs -->
  </xsl:call-template>
+++++++ Snippet End +++++++
```

Action: Change lines 125 to 127 to the following snippet

```
+++++++ Snippet Start +++++++
  <xsl:call-template name="parseAndDisplayConciseHoldings">
    <xsl:with-param name="whichPair" select="'4'"/>  <!-- display 854 / 864 pairs -->
  </xsl:call-template>
+++++++ Snippet End +++++++
```

Action: Change lines 117 to 119 to the following snippet

```
+++++++ Snippet Start +++++++
<xsl:message>IN DISPLAY1041</xsl:message>
  <xsl:call-template name="parseAndDisplayConciseHoldings">
    <xsl:with-param name="whichPair" select="'3'"/>  <!-- display 853 / 863 pairs -->
  </xsl:call-template>
+++++++ Snippet End +++++++
```

Action: Change line 100 to the following snippet

```
+++++++ Snippet Start +++++++
  <xsl:value-of select="$v104Xconfig//Config/IndexCaption"/>
+++++++ Snippet End +++++++
```

Action: Change line 96 to the following snippet

```
+++++++ Snippet Start +++++++
  <xsl:value-of select="$v104Xconfig//Config/SupplementCaption"/>
+++++++ Snippet End +++++++
```

Action: Change lines 83 to 90 to the following snippet

```
+++++++ Snippet Start +++++++
  <xsl:choose>
```

```
<xsl:when test="$displayFormat='STANDARD'">
  <xsl:value-of select="$v104Xconfig//Config/StandardChronologyRangeSeparator"/>
</xsl:when>
<xsl:when test="translate($displayFormat, $lowercase, $uppercase)='NISO'">
  <xsl:value-of select="$v104Xconfig//Config/NISOChronologyRangeSeparator"/>
</xsl:when>
</xsl:choose>
+++++++ Snippet End +++++++
```

Action: Change lines 72 to 79 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:choose>
  <xsl:when test="$displayFormat='STANDARD'">
    <xsl:value-of select="$v104Xconfig//Config/StandardChronologyLevelSeparator"/>
  </xsl:when>
  <xsl:when test="translate($displayFormat, $lowercase, $uppercase)='NISO'">
    <xsl:value-of select="$v104Xconfig//Config/NISOChronologyLevelSeparator"/>
  </xsl:when>
</xsl:choose>
+++++++ Snippet End ++++++++
```

Action: Change lines 61 to 68 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:choose>
  <xsl:when test="$displayFormat='STANDARD'">
    <xsl:value-of select="$v104Xconfig//Config/StandardChronLabelDataSeparator"/>
  </xsl:when>
  <xsl:when test="translate($displayFormat, $lowercase, $uppercase)='NISO'">
    <xsl:value-of select="$v104Xconfig//Config/NISOChronLabelDataSeparator"/>
  </xsl:when>
</xsl:choose>
+++++++ Snippet End ++++++++
```

Action: Change lines 50 to 57 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:choose>
  <xsl:when test="$displayFormat='STANDARD'">
    <xsl:value-of select="$v104Xconfig//Config/StandardEnumLabelDataSeparator"/>
  </xsl:when>
  <xsl:when test="translate($displayFormat, $lowercase, $uppercase)='NISO'">
    <xsl:value-of select="$v104Xconfig//Config/NISOEnumLabelDataSeparator"/>
  </xsl:when>
</xsl:choose>
+++++++ Snippet End ++++++++
```

Action: Change lines 39 to 46 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:choose>
  <xsl:when test="$displayFormat='STANDARD'">
    <xsl:value-of select="$v104Xconfig//Config/StandardEnumLevelSeparator"/>
  </xsl:when>
  <xsl:when test="$displayFormat='NISO'">
    <xsl:value-of select="$v104Xconfig//Config/NISOEnumLevelSeparator"/>
  </xsl:when>
</xsl:choose>
+++++++ Snippet End ++++++++
```

Action: Change line 22 to the following snippet

```
+++++++ Snippet Start ++++++++
  xmlns:slim="http://www.loc.gov/MARC21/slim">
+++++++ Snippet End ++++++++
```


Action: Change line 18 to the following snippet

+++++++ Snippet Start +++++++

** This file is based upon XSL coding created by the University of Kansas

+++++++ Snippet End +++++++

Action: Change lines 1 to 2 to the following snippet

+++++++ Snippet Start +++++++

<!-- \$Revision: 1.3.2.1.2.1.2.1 \$ \$Date: 2013/05/10 16:52:34 \$ -->

<!--

+++++++ Snippet End +++++++

/m1/voyager/xxxdb/tomcat/vxws/context/vxws/xsl/configs/OPAC_in_primo_104X_display.xsl

Action: Add the following snippet at starting at line 1658

+++++++ Snippet Start +++++++

```
<xsl:template name="countHyphenInSubfield">
```

```
  <xsl:param name="subfield"/>
```

```
  <xsl:variable name="hyphenCounter">
```

```
    <!-- Remove hyphens from string and check the difference to get the counter. -->
```

```
    <xsl:value-of select="string-length($subfield) - string-length(translate($subfield, '-', ''))"/>
```

```
  </xsl:variable>
```

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

```
</xsl:template>
```

```
<!-- ***** -->
```

```
<xsl:template name="getTextAfterSecondHyphen">
```

```
  <xsl:param name="text"/>
```

```
  <xsl:variable name="textAfterHyphen">
```

```
    <xsl:value-of select="substring-after($text, '-')"/>
```

```
  </xsl:variable>
```

```
  <xsl:value-of select="substring-after($textAfterHyphen, '-')"/>
```

```
</xsl:template>
```

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 1496

+++++++ Snippet Start +++++++

+++++++ Snippet End +++++++

Action: Add the following snippet at starting at line 1492

+++++++ Snippet Start +++++++

```
  <!-- Is there a hyphen? There is nothing after the hyphen but include hyphen in display. -->
```

```
  <xsl:when test="$hyphenCount = 1">
```

```
    <xsl:value-of select="concat(substring-before($text, '-'), '-'"/>
```

```
  </xsl:when>
```

+++++++ Snippet End +++++++

Action: Change lines 1489 to 1490 to the following snippet

+++++++ Snippet Start +++++++

```
  <xsl:variable name="hyphenCount">
```

```
    <xsl:call-template name="countHyphenInSubfield">
```

```
      <xsl:with-param name="subfield" select="$text"/>
```

```
    </xsl:call-template>
```

```
</xsl:variable>
```

```
<xsl:choose>
```

```
<!-- Expecting three hyphens if there is more than 1 -->
```

```
<xsl:when test="$hyphenCount > 1">
```

```
  <xsl:variable name="textAfterSecondHyphen">
```

```
    <xsl:call-template name="getTextAfterSecondHyphen">
```

```
      <xsl:with-param name="text" select="$text"/>
```

```
    </xsl:call-template>
```

```
  </xsl:variable>
```

```
<!-- Add a hyphen before the end part -->
```

```
<xsl:variable name="textAfterSecondHyphenWithPreHyphen">
```

```
  <xsl:value-of select="concat('-', $textAfterSecondHyphen)"/>
```

```
</xsl:variable>
```

```
  <xsl:value-of select="substring-before($text, $textAfterSecondHyphenWithPreHyphen)"/>
```

```
</xsl:when>
```

```
<!-- Is there a hyphen and something after the hyphen? -->
```

```
<xsl:when test="$hyphenCount = 1 and substring-after($text, '-')">
```

```
+++++++ Snippet End +++++++
```

```
#### Action: Change line 1357 to the following snippet ####
```

```
+++++++ Snippet Start +++++++
```

```
<xsl:if test="string-length($display)">
```

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

```
</xsl:if>
```

```
+++++++ Snippet End +++++++
```

```
#### Action: Change lines 1351 to 1355 to the following snippet ####
```

```
+++++++ Snippet Start +++++++
```

```
<xsl:choose>
```

```
  <xsl:when test="$hyphenCount = '1' or $hyphenCount != '3'">
```

```
    <xsl:call-template name="getChron">
```

```
      <xsl:with-param name="caption" select="$subfield5j"/>
```

```
      <xsl:with-param name="text" select="substring-after($subfield6j, '-')"/>
```

```
      <xsl:with-param name="lang" select="$lang"/>
```

```
    </xsl:call-template>
```

```
  </xsl:when>
```

```
<xsl:otherwise>
```

```
  <xsl:variable name="textAfterSecondHyphen">
```

```
    <xsl:call-template name="getTextAfterSecondHyphen">
```

```
      <xsl:with-param name="text" select="$subfield6j"/>
```

```
    </xsl:call-template>
```

```
  </xsl:variable>
```

```
  <xsl:call-template name="getChron">
```

```
    <xsl:with-param name="caption" select="$subfield5j"/>
```

```
    <xsl:with-param name="text" select="$textAfterSecondHyphen"/>
```

```
    <xsl:with-param name="lang" select="$lang"/>
```

```
  </xsl:call-template>
```

```
</xsl:otherwise>
```

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

```
+++++++ Snippet End +++++++
```

```
#### Action: Add the following snippet at starting at line 1349 ####
```

```
+++++++ Snippet Start +++++++
```

```
<!-- Count hyphens. Multiple dashes indicate a quarterly issue chronology -->
```

```

    <xsl:variable name="hyphenCount">
      <xsl:call-template name="countHyphenInSubfield">
        <xsl:with-param name="subfield" select="$subfield6j"/>
      </xsl:call-template>
    </xsl:variable>
+++++++ Snippet End ++++++++

```

Action: Change lines 1334 to 1342 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:variable name="displayl">
    <xsl:choose>
      <xsl:when test="string-length($subfield5i) and contains($subfield6i, '-')">
        <xsl:call-template name="getChron">
          <xsl:with-param name="caption" select="$subfield5i"/>
          <xsl:with-param name="text" select="substring-after($subfield6i, '-')"/>
          <xsl:with-param name="lang" select="$lang"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = """/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>

  <xsl:if test="string-length($displayl)">
+++++++ Snippet End ++++++++

```

Action: Delete lines 412 - 549

Action: Change lines 393 to 404 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:value-of select="$displayCode"/>
  </subfield>
</xml-fragment>
+++++++ Snippet End ++++++++

```

Action: Change lines 388 to 391 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:value-of select=""9"/>
+++++++ Snippet End ++++++++

```

Action: Change line 386 to the following snippet

```

+++++++ Snippet Start ++++++++
  <xsl:choose>
    <!-- Creating an ends-with($field853863Output, ';') test. Not in 1.0 -->
    <xsl:when test="substring($field853863Output, string-length($field853863Output), 1) = ';'>
      <xsl:value-of select="substring($field853863Output, 1, string-length($field853863Output)-1)"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$field853863Output"/>
    </xsl:otherwise>
  </xsl:choose>
</subfield>

  <subfield>
+++++++ Snippet End ++++++++

```

Action: Delete lines 380 - 384

Action: Change lines 375 to 378 to the following snippet

```

+++++++ Snippet Start ++++++++

```

```

        <xsl:value-of select="a"/>
+++++++ Snippet End ++++++

#### Action: Change line 373 to the following snippet ####
+++++++ Snippet Start ++++++
        <subfield>
+++++++ Snippet End ++++++

#### Action: Change lines 367 to 370 to the following snippet ####
+++++++ Snippet Start ++++++
        <xsl:value-of select="$collapseTag"/>
+++++++ Snippet End ++++++

#### Action: Change lines 353 to 364 to the following snippet ####
+++++++ Snippet Start ++++++
        <xsl:attribute name="ind1">
            <xsl:value-of select="$ind1From85x" />
        </xsl:attribute>
        <xsl:attribute name="ind2">
            <xsl:value-of select="1" />
        </xsl:attribute>
+++++++ Snippet End ++++++

#### Action: Add the following snippet at starting at line 350 ####
+++++++ Snippet Start ++++++
        <xsl:variable name="collapseTag">
            <xsl:choose>
                <xsl:when test="$whichPair='3'">
                    <xsl:value-of select="866" />
                </xsl:when>
                <xsl:when test="$whichPair='4'">
                    <xsl:value-of select="867" />
                </xsl:when>
                <xsl:when test="$whichPair='5'">
                    <xsl:value-of select="868" />
                </xsl:when>
            </xsl:choose>
        </xsl:variable>

+++++++ Snippet End ++++++

#### Action: Add the following snippet at starting at line 275 ####
+++++++ Snippet Start ++++++
        <xsl:variable name="ind1From85x">
            <xsl:value-of select="@ind1"/>
        </xsl:variable>

+++++++ Snippet End ++++++

#### Action: Add the following snippet at starting at line 225 ####
+++++++ Snippet Start ++++++
        <xsl:variable name="displayCode">
            <xsl:choose>
                <xsl:when test="$whichPair='3'">
                    <xsl:value-of select="1040" />
                </xsl:when>
                <xsl:when test="$whichPair='4'">
                    <xsl:value-of select="1042" />
                </xsl:when>
                <xsl:when test="$whichPair='5'">
                    <xsl:value-of select="1044" />
            </xsl:choose>
        </xsl:variable>

```

```
</xsl:when>
</xsl:choose>
</xsl:variable>
```

+++++++ Snippet End +++++++

Action: Delete lines 26 - 28

Action: Delete line 13

Action: Change line 1 to the following snippet

```
+++++++ Snippet Start ++++++++
<!-- $Revision: 1.3.2.1.2.10.2.3 $ $Date: 2013/05/10 17:06:34 $ -->
+++++++ Snippet End ++++++++
```

</m1/voyager/xxxdb/tomcat/vxws/context/vxws/xsl/transformOPACInPrimoHoldings.xsl>

Action: Change lines 63 to 64 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:copy-of select="$compressedData"/><br/>
  </xsl:if>
+++++++ Snippet End ++++++++
```

Action: Change lines 54 to 59 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:variable name="compressedData">
    <xsl:for-each select="//mfhd:marcRecord">
      <xsl:if test="$iField='1040'">
        <xsl:call-template name="display1041"/>
      </xsl:if>
    </xsl:for-each>
+++++++ Snippet End ++++++++
```

Action: Change line 52 to the following snippet

```
+++++++ Snippet Start ++++++++
<xsl:template name="BMD1040">
+++++++ Snippet End ++++++++
```

Action: Delete lines 48 - 49

Action: Change lines 43 to 45 to the following snippet

```
+++++++ Snippet Start ++++++++
  <xsl:call-template name="BMD1040">
    <xsl:with-param name="iField" select="'1040'"/>
  </xsl:call-template>
+++++++ Snippet End ++++++++
```

+++++++ Snippet End ++++++++

Action: Change line 37 to the following snippet

```
+++++++ Snippet Start ++++++++
<!-- create an XML fragment to be imported into the data field XML structure of the MARC record -->
+++++++ Snippet End ++++++++
```

Action: Change line 2 to the following snippet

```
+++++++ Snippet Start ++++++++
<!-- $Revision: 1.3.2.1.2.3.2.2 $ $Date: 2013/04/23 21:16:17 $ -->
+++++++ Snippet End ++++++++
```