```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:odm="http://www.cdisc.org/ns/odm/v1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:def="http://www.cdisc.org/ns/def/v2.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:arm="http://www.cdisc.org/ns/arm/v1.0" xml:lang="en"
  exclude-result-prefixes="def xlink odm xsi arm">
  <xsl:output method="html" indent="yes" encoding="utf-8"
doctype-system="http://www.w3.org/TR/html4/strict.dtd"
    doctype-public="-//W3C//DTD HTML 4.01//EN" version="4.0"/>


  <!-- Methods will be displayed, unless the displayMethods parameter has a value
of 0.
       This parameter can be set in the XSLT processor.
        -->
  <xsl:param name="displayMethods"/>

  <!-- Comments will be displayed, unless the displayComments parameter has a value
of 0.
       This parameter can be set in the XSLT processor.
        -->
  <xsl:param name="displayComments"/>

  <!--
*************************************************************************************
********************** -->
  <!-- File:   define2-0-0.xsl
                              -->
  <!-- Description: This stylesheet works with the Define-XML 2.0.0 specification,
including the Analysis        -->
  <!--              Results Metadata v1.0 extension.
                              -->
  <!-- Author: Lex Jansen (CDISC XML Technologies Team, ADaM Metadata Team)
                              -->
  <!-- Date:   2013-03-04 (Original version)
                              -->
  <!-- Changes:
                              -->
  <!--   2015-01-16 - Added Study metadata display
                              -->
  <!--              - Improved Analysis Parameter(s) display
                              -->
  <!--   2014-08-29 - Added displayMethods parameter.
                              -->
  <!--              - Added link when href has a value in ExternalCodeList
(AppendixExternalCodeLists template). -->
  <!--              - Many improvements for linking to external PDF documents with
physical page references or   -->
```

```
  <!--                named destinations.
                              -->
  <!--   2013-12-12 - Fixed with non-existing CodeList being linked.
                              -->
  <!--   2013-08-10 - Fixed issue in value level where clause display.
                              -->
  <!--              - Removed Comment sorting.
                              -->
  <!--              - Added Analysis Results Metadata.
                              -->
  <!--   2013-04-24 - Fixed issue in displayISO8601 template when ItemDef/@Name has
length=1.                     -->
  <!--   2013-03-04 - Initial version.
                              -->
  <!--
                              -->
  <!--   The CDISC Define-XML standard does not dictate how a stylesheet should
display a Define-XML v2 file.    -->
  <!--   This example stylesheet can be altered to satisfy alternate visualization
needs.                         -->
  <!--
********************************************************************************
********************* -->

  <!-- Global Variables -->
  <xsl:variable name="g_stylesheetVersion" select="'2015-01-16'"/>

  <!-- XSLT 1.0 does not support the function 'upper-case()'
       so we need to use the 'translate() function, which uses the variables
$lowercase and $uppercase.
       Remark that this is not a XSLT problem, but a problem that browsers like IE
do still not support XSLT 2.0 yet -->
  <xsl:variable name="LOWERCASE" select="'abcdefghijklmnopqrstuvwxyz'"/>
  <xsl:variable name="UPPERCASE" select="'ABCDEFGHIJKLMNOPQRSTUVWXYZ'"/>

  <xsl:variable name="REFTYPE_PHYSICALPAGE">PhysicalRef</xsl:variable>
  <xsl:variable name="REFTYPE_NAMEDDESTINATION">NamedDestination</xsl:variable>

  <xsl:variable name="g_StudyName"
select="/odm:ODM/odm:Study[1]/odm:GlobalVariables[1]/odm:StudyName"/>
  <xsl:variable name="g_StudyDescription"
select="/odm:ODM/odm:Study[1]/odm:GlobalVariables[1]/odm:StudyDescription"/>
  <xsl:variable name="g_ProtocolName"
select="/odm:ODM/odm:Study[1]/odm:GlobalVariables[1]/odm:ProtocolName"/>

  <xsl:variable name="g_MetaDataVersion"
select="/odm:ODM/odm:Study[1]/odm:MetaDataVersion[1]"/>
  <xsl:variable name="g_MetaDataVersionName" select="$g_MetaDataVersion/@Name"/>
  <xsl:variable name="g_MetaDataVersionDescription"
select="$g_MetaDataVersion/@Description"/>
```

```
  <xsl:variable name="g_DefineVersion"
select="$g_MetaDataVersion/@def:DefineVersion"/>
  <xsl:variable name="g_StandardName"
select="$g_MetaDataVersion/@def:StandardName"/>
  <xsl:variable name="g_StandardVersion"
select="$g_MetaDataVersion/@def:StandardVersion"/>

  <xsl:variable name="g_seqItemGroupDefs"
select="$g_MetaDataVersion/odm:ItemGroupDef"/>
  <xsl:variable name="g_seqItemDefs" select="$g_MetaDataVersion/odm:ItemDef"/>
  <xsl:variable name="g_seqCodeLists" select="$g_MetaDataVersion/odm:CodeList"/>
  <xsl:variable name="g_seqValueListDefs"
select="$g_MetaDataVersion/def:ValueListDef"/>
  <xsl:variable name="g_seqMethodDefs" select="$g_MetaDataVersion/odm:MethodDef"/>
  <xsl:variable name="g_seqCommentDefs"
select="$g_MetaDataVersion/def:CommentDef"/>
  <xsl:variable name="g_seqWhereClauseDefs"
select="$g_MetaDataVersion/def:WhereClauseDef"/>
  <xsl:variable name="g_seqleafs" select="$g_MetaDataVersion/def:leaf"/>

  <!--We need to be able to distuinguish between Tabulation and Analysis datasets
-->
  <xsl:variable name="g_nItemGroupDefs" select="count($g_seqItemGroupDefs)"/>
  <xsl:variable name="g_nItemGroupDefsAnalysis"
select="count($g_seqItemGroupDefs[@Purpose='Analysis'])"/>
  <xsl:variable name="g_nItemGroupDefsTabulation"
select="count($g_seqItemGroupDefs[@Purpose='Tabulation'])"/>
  <xsl:variable name="g_ItemGroupDefPurpose">
    <xsl:choose>
      <xsl:when test="($g_nItemGroupDefsAnalysis = $g_nItemGroupDefs) or
($g_nItemGroupDefsTabulation = $g_nItemGroupDefs)">
        <xsl:choose>
          <xsl:when test="($g_nItemGroupDefsTabulation = $g_nItemGroupDefs)">
            <xsl:text>Tabulation</xsl:text>
          </xsl:when>
          <xsl:when test="($g_nItemGroupDefsAnalysis = $g_nItemGroupDefs)">
            <xsl:text>Analysis</xsl:text>
          </xsl:when>
          <xsl:otherwise />
        </xsl:choose>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text></xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>


  <!-- ************************************************************** -->
  <!-- Create the HTML Header                                         -->
```

```
  <!-- ************************************************************ -->
  <xsl:template match="/">
    <html lang="en">
      <head>
        <meta http-equiv="Content-Script-Type" content="text/javascript"/>
        <meta http-equiv="Content-Style-Type" content="text/css"/>
        <title> Study <xsl:value-of
select="/odm:ODM/odm:Study/odm:GlobalVariables/odm:StudyName"/>, Data
Definitions</title>

        <xsl:call-template name="GenerateJavaScript"/>
        <xsl:call-template name="GenerateCSS"/>

      </head>
      <body onload="reset_menus();">
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="/odm:ODM/odm:Study/odm:GlobalVariables"/>
  <xsl:template match="/odm:ODM/odm:Study/odm:MetaDataVersion">

    <div id="menu">
      <!--  Skip Navigation Link for Accessibility -->
      <a name="top" class="invisible" href="#main">Skip Navigation Link</a>

      <span class="standard">
        <xsl:value-of select="$g_StandardName"/>
        <xsl:text> </xsl:text>
        <xsl:value-of select="$g_StandardVersion"/>
      </span>

      <ul class="hmenu">

        <!-- ************************************************* -->
        <!-- *************  Annotated CRF    ****************** -->
        <!-- ************************************************* -->
        <xsl:if test="$g_MetaDataVersion/def:AnnotatedCRF">
          <xsl:for-each
select="$g_MetaDataVersion/def:AnnotatedCRF/def:DocumentRef">
            <li class="hmenu-item">
              <span class="hmenu-bullet">+</span>
              <xsl:variable name="leafIDs" select="@leafID"/>
              <xsl:variable name="leaf" select="../../def:leaf[@ID=$leafIDs]"/>
              <a class="tocItem">
                <xsl:attribute name="href"><xsl:value-of
select="$leaf/@xlink:href"/></xsl:attribute>
                <xsl:value-of select="$leaf/def:title"/>
              </a>
```

```
            </li>
          </xsl:for-each>
        </xsl:if>

        <!-- ************************************************** -->
        <!-- *************  Supplemental Doc ****************** -->
        <!-- ************************************************** -->
        <xsl:if test="$g_MetaDataVersion/def:SupplementalDoc">
          <xsl:for-each
select="$g_MetaDataVersion/def:SupplementalDoc/def:DocumentRef">
            <li class="hmenu-item">
              <span class="hmenu-bullet">+</span>
              <xsl:variable name="leafIDs" select="@leafID"/>
              <xsl:variable name="leaf" select="../../def:leaf[@ID=$leafIDs]"/>
              <a class="tocItem">
                <xsl:attribute name="href"><xsl:value-of
select="$leaf/@xlink:href"/></xsl:attribute>
                <xsl:value-of select="$leaf/def:title"/>
              </a>
            </li>
          </xsl:for-each>
        </xsl:if>

        <!-- ************************************************** -->
        <!-- ************ Analysis Results Metadata ************* -->
        <!-- ************************************************** -->

        <xsl:if
test="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays">
                <li class="hmenu-submenu" >
                        <span class="hmenu-bullet"
onclick="toggle_submenu(this);">+</span>
                        <a class="tocItem" href="#ARM_Table_Summary" >Analysis
Results Metadata</a>
                        <ul>
                                <xsl:for-each
select="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays/arm:Resul
tDisplay">
                                        <li class="hmenu-item">
                                                <span class="hmenu-bullet">-</span>
                                                <a class="tocItem">
                                                  <xsl:attribute
name="href">#ARD.<xsl:value-of select="@OID"/></xsl:attribute>
                                                  <xsl:attribute
name="title"><xsl:value-of
select="./odm:Description/odm:TranslatedText"/></xsl:attribute>
                                                  <xsl:value-of select="@Name"/>
                                                </a>
                                        </li>
                                </xsl:for-each>
```

```
                    </ul>
                </li>
        </xsl:if>

        <!-- ************************************************** -->
        <!-- ************** Datasets ************************** -->
        <!-- ************************************************** -->
        <li class="hmenu-submenu">
          <span class="hmenu-bullet" onclick="toggle_submenu(this);">+</span>
          <a class="tocItem">
            <xsl:attribute name="href">
              <xsl:text>#</xsl:text><xsl:value-of select="$g_ItemGroupDefPurpose"
              /><xsl:text>_Datasets_Table</xsl:text>
            </xsl:attribute>
            <xsl:value-of select="$g_ItemGroupDefPurpose"/> Datasets</a>
          <ul>
            <xsl:for-each select="$g_seqItemGroupDefs">
              <li class="hmenu-item">
                <span class="hmenu-bullet">-</span>
                <a class="tocItem">
                  <xsl:attribute name="href">#IG.<xsl:value-of
select="@OID"/></xsl:attribute>
                  <xsl:choose>
                    <xsl:when test="@SASDatasetName">
                      <xsl:value-of
select="concat(./odm:Description/odm:TranslatedText, ' (', @SASDatasetName, ')')"/>
                    </xsl:when>
                    <xsl:otherwise>
                      <xsl:value-of
select="concat(./odm:Description/odm:TranslatedText, ' (', @Name, ')')"/>
                    </xsl:otherwise>
                  </xsl:choose>
                </a>
              </li>

            </xsl:for-each>
          </ul>
        </li>

        <!-- ************************************************** -->
        <!-- *************** Value Lists ********************** -->
        <!-- ************************************************** -->
        <xsl:if test="$g_seqValueListDefs">
          <li class="hmenu-submenu">
            <span class="hmenu-bullet" onclick="toggle_submenu(this);">+</span>

            <xsl:choose>
              <xsl:when test="$g_ItemGroupDefPurpose='Analysis'">
                <a class="tocItem" href="#valuemeta">Parameter Value Level
Metadata</a>
```

```
            </xsl:when>
            <xsl:when test="$g_ItemGroupDefPurpose='Tabulation'">
              <a class="tocItem" href="#valuemeta">Value Level Metadata</a>
            </xsl:when>
            <xsl:otherwise>
              <a class="tocItem" href="#valuemeta">Value Level Metadata</a>
            </xsl:otherwise>
          </xsl:choose>

          <ul>
            <xsl:for-each select="$g_seqValueListDefs">
              <li class="hmenu-item">
                <span class="hmenu-bullet">-</span>
                <!--  <a class="tocItem">-->

                <xsl:variable name="valueListDefOID" select="@OID"/>
                <xsl:variable name="valueListRef"

select="//odm:ItemDef/def:ValueListRef[@ValueListOID=$valueListDefOID]"/>
                <xsl:variable name="itemDefOID" select="$valueListRef/../@OID"/>

                <xsl:element name="a">

                  <xsl:choose>
                    <xsl:when
test="//odm:ItemRef[@ItemOID=$itemDefOID]/../@Name">
                      <!-- ValueList attached to an ItemGroup Item -->
                      <xsl:attribute name="class">tocItem</xsl:attribute>
                    </xsl:when>
                    <xsl:otherwise>
                      <!-- ValueList attached to a ValueList Item -->
                      <xsl:attribute name="class">tocItem level2</xsl:attribute>
                    </xsl:otherwise>
                  </xsl:choose>

                  <xsl:attribute name="href">#VL.<xsl:value-of
select="@OID"/></xsl:attribute>

                  <xsl:choose>
                    <xsl:when
test="//odm:ItemRef[@ItemOID=$itemDefOID]/../@Name">
                      <!-- ValueList attached to an ItemGroup Item -->
                      <xsl:value-of
select="//odm:ItemRef[@ItemOID=$itemDefOID]/../@Name"/>
                    </xsl:when>
                    <xsl:otherwise>
                      <!-- ValueList attached to a ValueList Item -->
                      <xsl:value-of
select="//odm:ItemRef[@ItemOID=$itemDefOID]/../@OID"/>
                    </xsl:otherwise>
```

```
                    </xsl:choose>

                    <xsl:text> [</xsl:text>
                    <xsl:value-of select="$valueListRef/../@Name"/>
                    <xsl:text>]</xsl:text>
                  </xsl:element>
                </li>
              </xsl:for-each>
            </ul>
          </li>
        </xsl:if>

        <!-- ************************************************** -->
        <!-- ******************* Code Lists ******************* -->
        <!-- ************************************************** -->
        <xsl:if test="$g_seqCodeLists">
          <li class="hmenu-submenu">
            <span onclick="toggle_submenu(this);" class="hmenu-bullet">+</span>
            <a href="#decodelist" class="tocItem">Controlled Terminology</a>
            <ul>

              <xsl:if test="$g_seqCodeLists[odm:CodeListItem|odm:EnumeratedItem]">
                <li class="hmenu-submenu">
                  <span class="hmenu-bullet"
onclick="toggle_submenu(this);">+</span>
                  <a class="tocItem" href="#decodelist">Controlled Terms</a>
                  <ul>
                    <xsl:for-each
select="$g_seqCodeLists[odm:CodeListItem|odm:EnumeratedItem]">
                      <li class="hmenu-item">
                        <span class="hmenu-bullet">-</span>
                        <a class="tocItem">
                          <xsl:attribute name="href">#CL.<xsl:value-of
select="@OID"/></xsl:attribute>
                          <xsl:value-of select="@Name"/>
                        </a>
                      </li>
                    </xsl:for-each>
                  </ul>
                </li>
              </xsl:if>

              <!-- ************************************************** -->
              <!-- ************** External Dictionaries *************** -->
              <!-- ************************************************** -->
              <xsl:if test="$g_seqCodeLists[odm:ExternalCodeList]">
                <li class="hmenu-submenu">
                  <span class="hmenu-bullet"
onclick="toggle_submenu(this);">+</span>
                  <a class="tocItem" href="#externaldictionary">External
```

```
Dictionaries</a>
                  <ul>
                    <xsl:for-each select="$g_seqCodeLists[odm:ExternalCodeList]">
                      <li class="hmenu-item">
                        <span class="hmenu-bullet">-</span>
                        <a class="tocItem">
                          <xsl:attribute name="href">#CL.<xsl:value-of
select="@OID"/></xsl:attribute>
                          <xsl:value-of select="@Name"/>
                        </a>
                      </li>
                    </xsl:for-each>
                  </ul>
                </li>
              </xsl:if>

            </ul>
          </li>

        </xsl:if>

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

        <xsl:if test="$displayMethods != '0'">
          <xsl:if test="$g_seqMethodDefs">
            <li class="hmenu-submenu">
              <span class="hmenu-bullet" onclick="toggle_submenu(this);">+</span>
              <a class="tocItem" href="#compmethod">
                <xsl:choose>
                  <xsl:when test="$g_ItemGroupDefPurpose='Tabulation'">
                    <xsl:text>Computational Algorithms</xsl:text>
                  </xsl:when>
                  <xsl:when test="$g_ItemGroupDefPurpose='Analysis'">
                    <xsl:text>Analysis Derivations</xsl:text>
                  </xsl:when>
                  <xsl:otherwise>Methods</xsl:otherwise>
                </xsl:choose>
              </a>
              <ul>
                <xsl:for-each select="$g_seqMethodDefs">
                  <li class="hmenu-item">
                    <span class="hmenu-bullet">-</span>
                    <a class="tocItem">
                      <xsl:attribute name="href">#MT.<xsl:value-of
select="@OID"/></xsl:attribute>
                      <xsl:value-of select="@Name"/>
                    </a>
                  </li>
```

```
              </xsl:for-each>
            </ul>
          </li>
        </xsl:if>
      </xsl:if>

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

      <xsl:if test="$displayComments != '0'">
        <xsl:if test="$g_seqCommentDefs">
          <li class="hmenu-submenu">
            <span class="hmenu-bullet" onclick="toggle_submenu(this);">+</span>
            <a class="tocItem" href="#comment">Comments</a>
            <ul>
              <xsl:for-each select="$g_seqCommentDefs">
                <li class="hmenu-item">
                  <span class="hmenu-bullet">-</span>
                  <a class="tocItem">
                    <xsl:attribute name="href">#COMM.<xsl:value-of
select="@OID"/></xsl:attribute>
                    <xsl:value-of select="@OID"/>
                  </a>
                </li>
              </xsl:for-each>
            </ul>
          </li>
        </xsl:if>
      </xsl:if>

    </ul>
  </div>
   <!-- end of menu -->



   <!-- start of main -->
   <div id="main">

     <div class="docinfo">
       <xsl:call-template name="DocGenerationDate"/>
       <xsl:call-template name="StylesheetDate"/>
     </div>


     <!-- Display a red banner in case this file does not contain only Tabulation
datasets or Analysis datasets -->
     <xsl:if test="($g_nItemGroupDefsTabulation &lt; $g_nItemGroupDefs) and
($g_nItemGroupDefsAnalysis &lt; $g_nItemGroupDefs)) or
```

```
        (($g_nItemGroupDefsTabulation &gt; 0) and ($g_nItemGroupDefsAnalysis &gt;
0))">
        <span class="error">It is expected that all ItemGroups have
Purpose='Tabulation' or all ItemGroups have Purpose='Analysis'.</span>
      </xsl:if>

      <!-- Display Study metadata -->
      <div class="study">
        <dl class="multiple-table">
          <dt>Standard</dt><dd><xsl:value-of
select="$g_StandardName"/> <xsl:value-of select="$g_StandardVersion"/></dd>
          <dt>Study Name</dt><dd><xsl:value-of select="$g_StudyName"/></dd>
          <dt>Study Description</dt><dd><xsl:value-of
select="$g_StudyDescription"/></dd>
          <dt>Protocol Name</dt><dd><xsl:value-of select="$g_ProtocolName"/></dd>
          <dt>Metadata Name</dt><dd><xsl:value-of
select="$g_MetaDataVersionName"/></dd>
          <xsl:if test="$g_MetaDataVersionDescription">
            <dt>Metadata Description</dt><dd><xsl:value-of
select="$g_MetaDataVersionDescription"/></dd>
          </xsl:if>
        </dl>
      </div>

      <!-- ************************************************************** -->
        <!-- Create the ADaM Results Metadata Tables                      -->
        <!-- ************************************************************** -->

        <xsl:if
test="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays">

              <xsl:call-template name="AnalysisResultsSummary"/>
          <xsl:call-template name="lineBreak"/>


              <!--
********************************************************** -->
              <!-- Create the ADaM Results Metadata Detail Tables
   -->
              <!--
********************************************************** -->

              <xsl:call-template name="AnalysisResultsDetails"/>
          <xsl:call-template name="lineBreak"/>

        </xsl:if>

        <!-- ************************************************************** -->
      <!-- Create the Data Definition Tables                              -->
      <!-- ************************************************************** -->
```

```
      <a id="{$g_ItemGroupDefPurpose}_Datasets_Table"/>

      <h1 class="invisible"><xsl:value-of select="$g_ItemGroupDefPurpose"/>
Datasets for Study <xsl:value-of
        select="/odm:ODM/odm:Study/odm:GlobalVariables/odm:StudyName"/>
(<xsl:value-of
          select="$g_StandardName"/>
        <xsl:text> </xsl:text>
        <xsl:value-of select="$g_StandardVersion"/>)</h1>
      <div class="containerbox" style="page-break-after: always;">

        <table summary="Data Definition Tables">
          <caption class="header"><xsl:value-of select="$g_ItemGroupDefPurpose"/>
Datasets for Study <xsl:value-of
              select="/odm:ODM/odm:Study/odm:GlobalVariables/odm:StudyName"/>
(<xsl:value-of
                select="$g_StandardName"/>
            <xsl:text> </xsl:text>
            <xsl:value-of select="$g_StandardVersion"/>)</caption>
          <tr class="header">
            <th scope="col">Dataset</th>
            <th scope="col">Description</th>
            <th scope="col">Class</th>
            <th scope="col">Structure</th>
            <th scope="col">Purpose</th>
            <th scope="col">Keys</th>
            <th scope="col">Location</th>
            <th scope="col">Documentation</th>
          </tr>
          <xsl:for-each select="$g_seqItemGroupDefs">
            <xsl:call-template name="ItemGroupDefs"/>
          </xsl:for-each>
        </table>

      </div>

      <xsl:call-template name="linkTop"/>
      <xsl:call-template name="lineBreak"/>

      <!-- ************************************************************** -->
      <!-- Detail for the ADaM Data Definition Tables (Analysis)          -->
      <!-- ************************************************************** -->

      <xsl:for-each select="$g_seqItemGroupDefs[@Purpose='Analysis']">
        <xsl:call-template name="ItemRefADaM"/>
        <xsl:call-template name="linkTop"/>
        <xsl:call-template name="lineBreak"/>
      </xsl:for-each>
```

```
      <!-- ************************************************************** -->
      <!-- Detail for the SDTM/SEND Data Definition Tables (Tabulation)   -->
      <!-- This template will also be used for any ItemGroup that has a   -->
      <!-- Purpose attribute not equal to 'Analysis'                      -->
      <!-- ************************************************************** -->

      <xsl:for-each select="$g_seqItemGroupDefs[@Purpose!='Analysis']">
        <xsl:call-template name="ItemRefSDS"/>
        <xsl:call-template name="linkTop"/>
        <xsl:call-template name="lineBreak"/>
      </xsl:for-each>

      <!-- ************************************************  -->
      <!-- Create the Value Level Metadata (Value List)      -->
      <!-- ************************************************  -->
      <xsl:call-template name="AppendixValueList"/>

      <!-- ************************************************************** -->
      <!-- Create the Code Lists, Enumerated Items and External Dictionaries -->
      <!-- ************************************************************** -->
      <xsl:call-template name="AppendixCodeLists"/>
      <xsl:call-template name="AppendixExternalCodeLists"/>

      <!-- ************************************************************** -->
      <!-- Create the Derivations                                         -->
      <!-- ************************************************************** -->
      <xsl:if test="$displayMethods != '0'">
        <xsl:call-template name="AppendixMethods"/>
      </xsl:if>

      <!-- ************************************************************** -->
      <!-- Create the Comments                                            -->
      <!-- ************************************************************** -->
      <xsl:if test="$displayComments != '0'">
        <xsl:call-template name="AppendixComments"/>
      </xsl:if>

    </div>
    <!-- end of main -->

  </xsl:template>


  <!-- ************************************************ -->
  <!-- Analysis Results Summary                         -->
  <!-- ************************************************ -->
  <xsl:template name="AnalysisResultsSummary">
    <div class="containerbox" style="page-break-after: always;">
      <h1 id="ARM_Table_Summary">Analysis Results Metadata (Summary) for Study
<xsl:value-of select="/odm:ODM/odm:Study/odm:GlobalVariables/odm:StudyName"/></h1>
```

```
      <table class="arm-table" summary="Analysis Results Metadata (Summary)">
        <xsl:for-each
select="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays/arm:Resul
tDisplay">
          <tr>
            <td>
              <xsl:variable name="DisplayOID" select="@OID"/>
              <xsl:variable name="DisplayName" select="@Name"/>
              <xsl:variable name="Display"
select="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays/arm:Resul
tDisplay[@OID=$DisplayOID]"/>
              <a>
                <xsl:attribute name="href">#ARD.<xsl:value-of
select="$DisplayOID"/></xsl:attribute>
                <xsl:value-of select="$DisplayName"/>
              </a>
              <span class="title">
                <xsl:value-of select="./odm:Description/odm:TranslatedText"/>
              </span>
              <!-- if there is  more than one analysis result, list each linked to
the respective rows in the detail tables-->
              <xsl:for-each select="./arm:AnalysisResult">
                <xsl:variable name="AnalysisResultID" select="./@OID"/>
                <xsl:variable name="AnalysisResult"
select="$Display/arm:AnalysisResults[@OID=$AnalysisResultID]"/>
                <p class="summaryresult">
                  <a>
                    <xsl:attribute name="href">#AR.<xsl:value-of
select="$AnalysisResultID"/></xsl:attribute>
                    <xsl:value-of select="./odm:Description/odm:TranslatedText"/>
                  </a>
                </p>
              </xsl:for-each>
            </td>
          </tr>
        </xsl:for-each>
      </table>
    </div>
  </xsl:template>

  <!-- ********************************************** -->
  <!--  Analysis Results Details                      -->
  <!-- ********************************************** -->
  <xsl:template name="AnalysisResultsDetails">
      <h1>Analysis Results Metadata (Detail) for Study <xsl:value-of
select="/odm:ODM/odm:Study/odm:GlobalVariables/odm:StudyName"/></h1>

      <xsl:for-each
select="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays/arm:Resul
tDisplay">
```

```
        <div class="containerbox" style="page-break-after: always;">
          <xsl:variable name="DisplayOID" select="@OID"/>
          <xsl:variable name="DisplayName" select="@Name"/>
          <xsl:variable name="Display"
select="/odm:ODM/odm:Study/odm:MetaDataVersion/arm:AnalysisResultDisplays/arm:Resul
tDisplay[@OID=$DisplayOID]"/>

          <a>
            <xsl:attribute name="id">ARD.<xsl:value-of
select="$DisplayOID"/></xsl:attribute>
          </a>

          <xsl:element name="table">

            <xsl:attribute name="summary">Analysis Results Metadata for
<xsl:value-of select="$DisplayName"/> (detail)</xsl:attribute>
            <caption>
              <xsl:value-of select="$DisplayName"/>
            </caption>

            <tr>
              <th scope="col" class="label">Display</th>
              <th scope="col">

                <xsl:for-each select="def:DocumentRef">
                  <xsl:variable name="leafID" select="@leafID"/>
                  <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

                  <xsl:variable name="title" select="$leaf/def:title"/>
                  <xsl:variable name="href" select="$leaf/@xlink:href"/>
                  <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
                  <xsl:variable name="PageRefs"
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
                  <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
                  <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

                  <span class="linebreakcell">
                    <xsl:call-template name="createHyperLink">
                      <xsl:with-param name="href" select="$href"/>
                      <xsl:with-param name="PageRefType" select="$PageRefType"/>
                      <xsl:with-param name="PageRefs" select="$PageRefs"/>
                      <xsl:with-param name="PageFirst" select="$PageFirst"/>
                      <xsl:with-param name="PageLast" select="$PageLast"/>
                      <xsl:with-param name="title" select="$title"/>
                  </xsl:call-template>
                  </span>
```

```
                </xsl:for-each>
                <span class="title"><xsl:value-of
select="$Display/odm:Description/odm:TranslatedText"/></span>
              </th>
            </tr>

            <!--
                  Analysis Results
                -->

            <xsl:for-each select="$Display/arm:AnalysisResult">
              <xsl:variable name="AnalysisResultOID" select="@OID"/>
              <xsl:variable name="AnalysisResult"
select="$Display/arm:AnalysisResult[@OID=$AnalysisResultOID]"/>
              <tr class="analysisresult">
                <td>Analysis Result</td>
                <td>
                  <!--  add an identifier to Analysis Results xsl:value-of
select="OID"/-->
                  <span>
                    <xsl:attribute name="id">AR.<xsl:value-of
select="$AnalysisResultOID"/></xsl:attribute>
                    <xsl:value-of select="odm:Description/odm:TranslatedText"/>
                  </span>
                </td>
              </tr>

              <!--
                  Get the analysis parameter code from the where clause,
                  and then get the parameter from the decode in the codelist.
                -->

              <xsl:variable name="ParameterOID"
select="$AnalysisResult/@ParameterOID"/>
              <tr>

                <td class="label">Analysis Parameter(s)</td>

                <td>

                  <xsl:for-each
select="$AnalysisResult/arm:AnalysisDatasets/arm:AnalysisDataset">

                    <xsl:variable name="WhereClauseOID"
select="def:WhereClauseRef/@WhereClauseOID"/>
                    <xsl:variable name="WhereClauseDef"
select="$g_seqWhereClauseDefs[@OID=$WhereClauseOID]"/>
                    <xsl:variable name="ItemGroupOID" select="@ItemGroupOID"/>
```

```
                    <!--  Get the RangeCheck associated with the parameter
(typically only one ...) -->
                    <xsl:for-each
select="$WhereClauseDef/odm:RangeCheck[@def:ItemOID=$ParameterOID]">

                      <xsl:variable name="whereRefItemOID"
select="./@def:ItemOID"/>
                      <xsl:variable name="whereRefItemName"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@Name"/>
                      <xsl:variable name="whereRefItemDataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
                      <xsl:variable name="whereOP" select="./@Comparator"/>
                      <xsl:variable name="whereRefItemCodeListOID"

select="$g_seqItemDefs[@OID=$whereRefItemOID]/odm:CodeListRef/@CodeListOID"/>
                      <xsl:variable name="whereRefItemCodeList"
                        select="$g_seqCodeLists[@OID=$whereRefItemCodeListOID]"/>

                      <xsl:call-template name="linkItemGroupItem">
                        <xsl:with-param name="ItemGroupOID"
select="$ItemGroupOID"/>
                        <xsl:with-param name="ItemOID" select="$whereRefItemOID"/>
                        <xsl:with-param name="ItemName"
select="$whereRefItemName"/>
                      </xsl:call-template>

                      <xsl:choose>
                        <xsl:when test="$whereOP = 'IN' or $whereOP = 'NOTIN'">
                          <xsl:text> </xsl:text>
                          <xsl:variable name="Nvalues"
select="count(./odm:CheckValue)"/>
                          <xsl:choose>
                            <xsl:when test="$whereOP='IN'">
                              <xsl:text> IN </xsl:text>
                            </xsl:when>
                            <xsl:otherwise>
                              <xsl:text> NOT IN </xsl:text>
                            </xsl:otherwise>
                          </xsl:choose>
                          <xsl:text> (</xsl:text>
                          <xsl:for-each select="./odm:CheckValue">
                            <xsl:variable name="CheckValueINNOTIN" select="."/>
                            <p class="linebreakcell">
                              <xsl:call-template name="displayValue">
                                <xsl:with-param name="Value"
select="$CheckValueINNOTIN"/>
                                <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
                                <xsl:with-param name="decode" select="1"/>
                                <xsl:with-param name="CodeList"
```

```
select="$whereRefItemCodeList"/>
                              </xsl:call-template>
                              <xsl:if test="position() != $Nvalues">
                                <xsl:value-of select="', '"/>
                              </xsl:if>
                            </p>
                          </xsl:for-each><xsl:text> ) </xsl:text>
                        </xsl:when>

                        <xsl:when test="$whereOP = 'EQ'">
                          <xsl:variable name="CheckValueEQ"
select="./odm:CheckValue"/>
                          <xsl:text> = </xsl:text>
                          <xsl:call-template name="displayValue">
                            <xsl:with-param name="Value" select="$CheckValueEQ"/>
                            <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
                            <xsl:with-param name="decode" select="1"/>
                            <xsl:with-param name="CodeList"
select="$whereRefItemCodeList"/>
                          </xsl:call-template>
                        </xsl:when>

                        <xsl:when test="$whereOP = 'NE'">
                          <xsl:variable name="CheckValueNE"
select="./odm:CheckValue"/>
                          <xsl:text> &#x2260; </xsl:text>
                          <xsl:call-template name="displayValue">
                            <xsl:with-param name="Value" select="$CheckValueNE"/>
                            <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
                            <xsl:with-param name="decode" select="1"/>
                            <xsl:with-param name="CodeList"
select="$whereRefItemCodeList"/>
                          </xsl:call-template>
                        </xsl:when>

                        <xsl:otherwise>
                          <xsl:variable name="CheckValueOTH"
select="./odm:CheckValue"/>
                          <xsl:text> </xsl:text>
                          <xsl:choose>
                            <xsl:when test="$whereOP='LT'">
                              <xsl:text> &lt; </xsl:text>
                            </xsl:when>
                            <xsl:when test="$whereOP='LE'">
                              <xsl:text> &lt;= </xsl:text>
                            </xsl:when>
                            <xsl:when test="$whereOP='GT'">
                              <xsl:text> &gt; </xsl:text>
```

```
                            </xsl:when>
                            <xsl:when test="$whereOP='GE'">
                              <xsl:text> &gt;= </xsl:text>
                            </xsl:when>
                            <xsl:otherwise>
                              <xsl:value-of select="$whereOP"/>
                            </xsl:otherwise>
                          </xsl:choose>
                          <xsl:call-template name="displayValue">
                            <xsl:with-param name="Value" select="$CheckValueOTH"/>
                            <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
                            <xsl:with-param name="decode" select="1"/>
                            <xsl:with-param name="CodeList"
select="$whereRefItemCodeList"/>
                          </xsl:call-template>
                        </xsl:otherwise>
                      </xsl:choose>

                      <br/>
                      <xsl:if test="position() != last()">
                        <xsl:text> and </xsl:text>
                      </xsl:if>

                    </xsl:for-each>

                    <!--  END - Get the RangeCheck associated with the parameter
(typically only one ...) -->

                  </xsl:for-each>

                </td>
              </tr>

              <!--
                  The analysis Variables are next. It will link to ItemDef
information.
                -->
              <tr>
                <td class="label">Analysis Variable(s)</td>
                <td>
                  <xsl:for-each select="arm:AnalysisDatasets/arm:AnalysisDataset">
                    <xsl:variable name="ItemGroupOID" select="@ItemGroupOID"/>
                    <xsl:for-each select="arm:AnalysisVariable">
                      <xsl:variable name="ItemOID" select="@ItemOID"/>
                      <xsl:variable name="ItemDef"
select="/odm:ODM/odm:Study/odm:MetaDataVersion/odm:ItemDef[@OID=$ItemOID]"/>
                        <p class="analysisvariable">
                        <a>
                          <xsl:attribute name="href">#<xsl:value-of
```

```
select="$ItemGroupOID"/>.<xsl:value-of select="$ItemOID"/></xsl:attribute>
                          <xsl:value-of select="$ItemDef/@Name"/>
                        </a> (<xsl:value-of
select="$ItemDef/odm:Description/odm:TranslatedText"/>)
                      </p>
                    </xsl:for-each>
                  </xsl:for-each>
                </td>

              </tr>

              <!-- Use the AnalysisReason attribute of the AnalysisResults -->
              <tr>
                <td class="label">Analysis Reason</td>
                <td><xsl:value-of select="$AnalysisResult/@AnalysisReason"/></td>
              </tr>
              <!-- Use the AnalysisPurpose attribute of the AnalysisResults -->
              <tr>
                <td class="label">Analysis Purpose</td>
                <td><xsl:value-of select="$AnalysisResult/@AnalysisPurpose"/></td>
              </tr>

              <!--
                  AnalysisDataset Data References
                -->
              <tr>
                <td class="label">Data References (incl. Selection Criteria)</td>
                <td>
                  <xsl:for-each
select="$AnalysisResult/arm:AnalysisDatasets/arm:AnalysisDataset">
                    <xsl:variable name="ItemGroupOID" select="@ItemGroupOID"/>
                    <xsl:variable name="ItemGroupDef"
select="/odm:ODM/odm:Study/odm:MetaDataVersion/odm:ItemGroupDef[@OID=$ItemGroupOID]
"/>
                    <div class="datareference">
                      <a>
                        <xsl:attribute name="href">#<xsl:value-of
select="$ItemGroupDef/@OID"/></xsl:attribute>
                        <xsl:attribute name="title"><xsl:value-of
select="$ItemGroupDef/odm:Description/odm:TranslatedText"/></xsl:attribute>
                        <xsl:value-of select="$ItemGroupDef/@Name"/>
                      </a>
                      <xsl:text>  [</xsl:text>
                      <xsl:call-template name="assembleWhereText">
                        <xsl:with-param name="ValueItemRef"

select="$AnalysisResult/arm:AnalysisDatasets/arm:AnalysisDataset[@ItemGroupOID=$Ite
mGroupOID]"/>
                        <xsl:with-param name="ItemGroupLink"
select="$ItemGroupOID"/>
```

```
                        <xsl:with-param name="decode" select="0"/>
                        <xsl:with-param name="break" select="0"/>
                      </xsl:call-template>
                      <xsl:text>]</xsl:text>
                      <xsl:variable name="whereclausecmntOID"
select="def:WhereClauseRef/@def:CommentOID"/>
                      <xsl:if test="$whereclausecmntOID">
                        <xsl:call-template name="displayItemDefComment">
                          <xsl:with-param name="itemDef"
select="$AnalysisResult/arm:AnalysisDatasets/arm:AnalysisDataset[@ItemGroupOID=$Ite
mGroupOID]/def:WhereClauseRef"/>
                        </xsl:call-template>
                      </xsl:if>
                    </div>

                  </xsl:for-each>

                  <!--AnalysisDatasets Comments-->
                  <xsl:for-each select="$AnalysisResult/arm:AnalysisDatasets">
                    <xsl:if test="@def:CommentOID">
                      <xsl:call-template name="displayItemGroupComment"/>
                    </xsl:if>
                  </xsl:for-each>

                 </td>
                </tr>

                <!--
                    if we have an arm:Documentation element
                    produce a row with the contained information
                  -->

                <xsl:for-each select="$AnalysisResult/arm:Documentation">
                  <tr>
                    <td class="label">Documentation</td>
                    <td>
                      <span>
                        <xsl:value-of
select="$AnalysisResult/arm:Documentation/odm:Description/odm:TranslatedText"/>
                      </span>

                      <xsl:for-each select="def:DocumentRef">
                        <xsl:variable name="leafID" select="@leafID"/>
                        <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

                        <xsl:variable name="title" select="$leaf/def:title"/>
                        <xsl:variable name="href" select="$leaf/@xlink:href"/>
                        <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
                        <xsl:variable name="PageRefs"
```

```
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
                    <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
                    <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

                    <p class="linebreakcell">
                     <xsl:call-template name="createHyperLink">
                       <xsl:with-param name="href" select="$href"/>
                       <xsl:with-param name="PageRefType" select="$PageRefType"/>
                       <xsl:with-param name="PageRefs" select="$PageRefs"/>
                       <xsl:with-param name="PageFirst" select="$PageFirst"/>
                       <xsl:with-param name="PageLast" select="$PageLast"/>
                       <xsl:with-param name="title" select="$title"/>
                     </xsl:call-template>
                    </p>

                  </xsl:for-each>
                </td>
              </tr>
            </xsl:for-each>

            <!--
                if we have a arm:ProgrammingCode element
                produce a row with the contained information
             -->
            <xsl:for-each select="$AnalysisResult/arm:ProgrammingCode">
              <tr>
                <td class="label">Programming Statements</td>
                <td>

                  <xsl:if test="@Context">
                      <span class="code-context">[<xsl:value-of
select="@Context"/>]</span>
                  </xsl:if>

                  <xsl:if test="arm:Code">
                    <pre class="code"><xsl:value-of select="arm:Code"/></pre>
                  </xsl:if>

                  <div class="code-ref">
                  <xsl:for-each select="def:DocumentRef">
                    <xsl:variable name="leafID" select="@leafID"/>
                    <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

                    <xsl:variable name="title" select="$leaf/def:title"/>
                    <xsl:variable name="href" select="$leaf/@xlink:href"/>
                    <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
                    <xsl:variable name="PageRefs"
```

```
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
                      <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
                      <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

                      <p class="linebreakcell">
                        <xsl:call-template name="createHyperLink">
                          <xsl:with-param name="href" select="$href"/>
                          <xsl:with-param name="PageRefType"
select="$PageRefType"/>
                          <xsl:with-param name="PageRefs" select="$PageRefs"/>
                          <xsl:with-param name="PageFirst" select="$PageFirst"/>
                          <xsl:with-param name="PageLast" select="$PageLast"/>
                          <xsl:with-param name="title" select="$title"/>
                        </xsl:call-template>
                      </p>

                    </xsl:for-each>
                    </div>
                  </td>
                </tr>
              </xsl:for-each>

            </xsl:for-each>
          </xsl:element>
        </div>

        <xsl:call-template name="linkTop"/>
        <xsl:call-template name="lineBreak"/>

      </xsl:for-each>
  </xsl:template>


  <!-- ************************************************* -->
  <!-- Template: ItemGroupDefs                           -->
  <!-- ************************************************* -->
  <xsl:template name="ItemGroupDefs">
    <xsl:param name="rowNum"/>

    <xsl:element name="tr">

      <xsl:call-template name="rowClass">
        <xsl:with-param name="rowNum" select="position()"/>
      </xsl:call-template>

      <!-- Create an anchor -->
      <xsl:attribute name="id">
        <xsl:value-of select="@OID"/>
```

```
      </xsl:attribute>

      <td>
        <xsl:value-of select="@Name"/>
      </td>

      <!-- ************************************************************ -->
      <!-- Link each ItemGroup to its corresponding section in the define  -->
      <!-- ************************************************************ -->
      <td>
        <a>
          <xsl:attribute name="href">#IG.<xsl:value-of
select="@OID"/></xsl:attribute>
          <xsl:value-of select="odm:Description/odm:TranslatedText"/>
        </a>
        <xsl:if test="odm:Alias[@Context='DomainDescription']">
          <xsl:text> (</xsl:text><xsl:value-of
select="odm:Alias/@Name"/><xsl:text>)</xsl:text>
        </xsl:if>
      </td>

      <td>
        <xsl:value-of select="@def:Class"/>
      </td>
      <td>
        <xsl:value-of select="@def:Structure"/>
      </td>
      <xsl:element name="td">
        <xsl:choose>
          <xsl:when test="@Purpose='Tabulation' or @Purpose='Analysis'">
             <xsl:value-of select="@Purpose"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:attribute name="class">error</xsl:attribute>
            <xsl:value-of select="@Purpose"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:element>
      <td>
        <xsl:call-template name="displayKeys"/>
      </td>

      <!-- ************************************************* -->
      <!-- Link each Dataset to its corresponding archive file  -->
      <!-- ************************************************* -->
      <td>
        <a>
          <xsl:attribute name="href">
            <xsl:value-of select="def:leaf/@xlink:href"/>
          </xsl:attribute>
```

```
          <xsl:value-of select="def:leaf/def:title"/>
        </a>
      </td>

      <!-- ********************************************** -->
      <!-- Comments                                       -->
      <!-- ********************************************** -->
      <td>
        <xsl:if test="@def:CommentOID">
          <xsl:call-template name="displayItemGroupComment"/>
        </xsl:if>
      </td>

    </xsl:element>
  </xsl:template>

  <!-- ************************************************** -->
  <!-- Template: ItemRefADaM (@Purpose='Analysis')        -->
  <!-- ************************************************** -->
  <xsl:template name="ItemRefADaM">

    <a id="IG.{@OID}"/>
    <div class="containerbox" style="page-break-after: always;">

      <h1 class="invisible">
        <xsl:choose>
          <xsl:when test="@SASDatasetName">
            <xsl:value-of select="concat(./odm:Description/odm:TranslatedText, '
(', @SASDatasetName, ') ')"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select="concat(./odm:Description/odm:TranslatedText, '
(', @Name, ') ')"/>
          </xsl:otherwise>
        </xsl:choose>
      </h1>

      <xsl:element name="table">
        <xsl:attribute name="summary">ItemGroup IG.<xsl:value-of select="@OID"/>
        </xsl:attribute>

        <caption>
          <xsl:call-template name="linkDataset"/>
        </caption>

        <!-- Output the column headers -->
        <tr class="header">
          <th scope="col">Variable</th>
          <th scope="col">Label</th>
                <th scope="col">Key</th>
```

```
            <th scope="col">Type</th>
          <th scope="col" class="length" abbr="Length">Length / Display Format</th>
          <th scope="col" abbr="Format">Controlled Terms or Format</th>
          <th scope="col" abbr="Derivation">Source/Derivation/Comment</th>
        </tr>
        <!-- Get the individual data points -->
        <xsl:for-each select="./odm:ItemRef">

          <xsl:sort data-type="number" order="ascending" select="@OrderNumber"/>
          <xsl:variable name="itemRef" select="."/>
          <xsl:variable name="itemDefOid" select="@ItemOID"/>
          <xsl:variable name="itemDef"
select="../../odm:ItemDef[@OID=$itemDefOid]"/>

          <xsl:element name="tr">

            <!-- Create an anchor -->
            <xsl:attribute name="id">
              <xsl:value-of select="$itemDef/@OID"/>
            </xsl:attribute>

            <xsl:call-template name="rowClass">
              <xsl:with-param name="rowNum" select="position()"/>
            </xsl:call-template>

            <td>
              <xsl:choose>
                <xsl:when test="$itemDef/def:ValueListRef/@ValueListOID!=''">
                  <a>
                    <xsl:attribute name="id">
                      <xsl:value-of select="../@OID"/>.<xsl:value-of
select="$itemDef/@OID"/>
                    </xsl:attribute>
                    <xsl:attribute name="href">#VL.<xsl:value-of
select="$itemDef/def:ValueListRef/@ValueListOID"/>
                    </xsl:attribute>
                    <xsl:attribute name="title">link to VL.<xsl:value-of
                        select="$itemDef/def:ValueListRef/@ValueListOID"/>
                    </xsl:attribute>
                    <xsl:value-of select="$itemDef/@Name"/>
                  </a>
                </xsl:when>
                <xsl:otherwise>
                  <!-- Make unique anchor link to Variable Name -->
                  <a>
                    <xsl:attribute name="name">
                      <xsl:value-of select="../@OID"/>.<xsl:value-of
select="$itemDef/@OID"/>
                    </xsl:attribute>
                  </a>
```

```
                <xsl:value-of select="$itemDef/@Name"/>
              </xsl:otherwise>
            </xsl:choose>
          </td>

          <td><xsl:value-of
select="$itemDef/odm:Description/odm:TranslatedText"/></td>
              <td class="number"><xsl:value-of select="@KeySequence"/></td>
          <td class="datatype"><xsl:value-of select="$itemDef/@DataType"/></td>

          <td class="number">
            <xsl:choose>
              <xsl:when test="$itemDef/@def:DisplayFormat">
                <xsl:value-of select="$itemDef/@def:DisplayFormat"/>
              </xsl:when>
              <xsl:otherwise>
                <xsl:value-of select="$itemDef/@Length"/>
              </xsl:otherwise>
            </xsl:choose>
          </td>

          <!-- ************************************************* -->
          <!-- Hypertext Link to the Decode Appendix             -->
          <!-- ************************************************* -->
          <td>
            <xsl:call-template name="linkDecodeList">
              <xsl:with-param name="itemDef" select="$itemDef"/>
            </xsl:call-template>

            <xsl:call-template name="displayISO8601">
              <xsl:with-param name="itemDef" select="$itemDef"/>
            </xsl:call-template>
          </td>

          <!-- ************************************************* -->
          <!--                  Comments Column                  -->
          <!-- ************************************************* -->
          <td class="linebreakcell">

            <xsl:variable name="Origin" select="$itemDef/def:Origin"/>

            <xsl:if test="$Origin">
              <xsl:choose>
                <xsl:when test="$Origin[@Type='Predecessor']">Predecessor:
<xsl:value-of

select="$itemDef/def:Origin/odm:Description/odm:TranslatedText"/>
                </xsl:when>
                <xsl:otherwise>
                  <xsl:value-of select="$Origin/@Type"/>
```

```
                <xsl:text>: </xsl:text>
              </xsl:otherwise>
            </xsl:choose>
          </xsl:if>


          <xsl:variable name="methodOID" select="$itemRef/@MethodOID"/>
          <xsl:if test="$methodOID">
            <xsl:call-template name="displayItemDefMethod">
              <xsl:with-param name="itemRef" select="$itemRef"/>
            </xsl:call-template>
          </xsl:if>

          <xsl:variable name="cmntOID" select="$itemDef/@def:CommentOID"/>
          <xsl:if test="$cmntOID">
            <xsl:call-template name="displayItemDefComment">
              <xsl:with-param name="itemDef" select="$itemDef"/>
            </xsl:call-template>
          </xsl:if>

        </td>

      </xsl:element>
    </xsl:for-each>
  </xsl:element>
 </div>
</xsl:template>

<!-- ********************************************************** -->
<!-- Template: ItemRefSDS (SDTM or SEND, (@Purpose!='Analysis'))  -->
<!-- ********************************************************** -->
<xsl:template name="ItemRefSDS">

  <a id="IG.{@OID}"/>
  <div class="containerbox" style="page-break-after: always;">

    <h1 class="invisible">
      <xsl:choose>
        <xsl:when test="@SASDatasetName">
          <xsl:value-of select="concat(./odm:Description/odm:TranslatedText, '
(', @SASDatasetName, ') ')"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="concat(./odm:Description/odm:TranslatedText, '
(', @Name, ') ')"/>
        </xsl:otherwise>
      </xsl:choose>
    </h1>

    <xsl:element name="table">
```

```
        <xsl:attribute name="summary">ItemGroup IG.<xsl:value-of select="@OID"/>
        </xsl:attribute>

        <caption>
          <xsl:call-template name="linkDataset"/>
        </caption>

        <!-- Output the column headers -->
        <tr class="header">
          <th scope="col">Variable</th>
          <th scope="col">Label</th>
          <th scope="col">Key</th>
          <th scope="col">Type</th>
          <th scope="col" class="length">Length</th>
          <th scope="col" abbr="Format">Controlled Terms or Format</th>
          <th scope="col">Origin</th>
          <th scope="col">Derivation/Comment</th>
        </tr>
        <!-- Get the individual data points -->
        <xsl:for-each select="./odm:ItemRef">

          <xsl:sort data-type="number" order="ascending" select="@OrderNumber"/>
          <xsl:variable name="itemRef" select="."/>
          <xsl:variable name="itemDefOid" select="@ItemOID"/>
          <xsl:variable name="itemDef"
select="../../odm:ItemDef[@OID=$itemDefOid]"/>

          <xsl:element name="tr">

            <xsl:call-template name="rowClass">
              <xsl:with-param name="rowNum" select="position()"/>
            </xsl:call-template>

            <td>
              <xsl:choose>
                <xsl:when test="$itemDef/def:ValueListRef/@ValueListOID!=''">
                  <a>
                    <xsl:attribute name="id">
                      <xsl:value-of select="../@OID"/>.<xsl:value-of
select="$itemDef/@OID"/>
                    </xsl:attribute>
                    <xsl:attribute name="href">#VL.<xsl:value-of
select="$itemDef/def:ValueListRef/@ValueListOID"/>
                    </xsl:attribute>
                    <xsl:attribute name="title">link to VL.<xsl:value-of
                        select="$itemDef/def:ValueListRef/@ValueListOID"/>
                    </xsl:attribute>
                    <xsl:value-of select="$itemDef/@Name"/>
                  </a>
                </xsl:when>
```

```
                <xsl:otherwise>
                  <!-- Make unique anchor link to Variable Name -->
                  <a>
                    <xsl:attribute name="name">
                      <xsl:value-of select="../@OID"/>.<xsl:value-of
select="$itemDef/@OID"/>
                    </xsl:attribute>
                  </a>
                  <xsl:value-of select="$itemDef/@Name"/>
                </xsl:otherwise>
              </xsl:choose>
            </td>

            <td><xsl:value-of
select="$itemDef/odm:Description/odm:TranslatedText"/></td>
            <td class="number"><xsl:value-of select="@KeySequence"/></td>
            <td class="datatype"><xsl:value-of select="$itemDef/@DataType"/></td>
            <td class="number"><xsl:value-of select="$itemDef/@Length"/></td>

            <!-- ************************************************ -->
            <!-- Hypertext Link to the Decode Appendix            -->
            <!-- ************************************************ -->
            <td>
              <xsl:call-template name="linkDecodeList">
                <xsl:with-param name="itemDef" select="$itemDef"/>
              </xsl:call-template>

              <xsl:call-template name="displayISO8601">
                <xsl:with-param name="itemDef" select="$itemDef"/>
              </xsl:call-template>
            </td>

            <!-- ************************************************ -->
            <!-- Origin Column for ItemDefs                       -->
            <!-- ************************************************ -->
            <td>
              <xsl:call-template name="displayItemDefOrigin">
                <xsl:with-param name="itemDef" select="$itemDef"/>
              </xsl:call-template>
            </td>

            <!-- ************************************************ -->
            <!-- Hypertext Link to the Derivation                 -->
            <!-- ************************************************ -->
            <td>

              <xsl:variable name="cmntOID" select="$itemDef/@def:CommentOID"/>
              <xsl:if test="$cmntOID">
                <xsl:call-template name="displayItemDefComment">
                  <xsl:with-param name="itemDef" select="$itemDef"/>
```

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

            <xsl:variable name="methodOID" select="$itemRef/@MethodOID"/>
            <xsl:if test="$methodOID">
              <xsl:call-template name="displayItemDefMethod">
                <xsl:with-param name="itemRef" select="$itemRef"/>
              </xsl:call-template>
            </xsl:if>

          </td>
        </xsl:element>
      </xsl:for-each>


      <!-- ************************************************ -->
      <!-- Link to SUPPXX domain                            -->
      <!-- For those domains with Suplemental Qualifiers    -->
      <!-- ************************************************ -->
      <!-- REMARK that we are still in the 'ItemRef' template
           but at the 'ItemGroupDef' level -->

      <xsl:variable name="datasetName" select="@Name"/>
      <xsl:variable name="suppDatasetName" select="concat('SUPP',$datasetName)"/>
      <xsl:if test="../odm:ItemGroupDef[@Name=$suppDatasetName]">
        <!-- create an extra row to the SUPPXX dataset when there is one -->
        <xsl:variable name="datasetOID"
select="../odm:ItemGroupDef[@Name=$suppDatasetName]"/>
        <tr>
          <td colspan="8">
            <xsl:text>Related dataset: </xsl:text>
            <xsl:value-of
select="../odm:ItemGroupDef[@Name=$suppDatasetName]/odm:Description/odm:TranslatedT
ext"/>
            <xsl:text> (</xsl:text>
            <a>
              <xsl:attribute name="href">#IG.<xsl:value-of
select="$datasetOID/@OID"/></xsl:attribute>
              <xsl:value-of select="$suppDatasetName"/>)</a>
          </td>
        </tr>
      </xsl:if>

      <!-- ************************************************ -->
      <!-- Link to Parent domain                            -->
      <!-- For those domains that are Suplemental Qualifiers   -->
      <!-- ************************************************ -->
      <!-- REMARK that we are still in the 'ItemRef' template
           but at the 'ItemGroupDef' level -->
```

```
        <xsl:if test="starts-with($datasetName, 'SUPP')">
          <!-- create an extra row to the XX dataset when there is one -->
          <xsl:variable name="parentDatasetName" select="substring($datasetName,
5)"/>
          <xsl:if test="../odm:ItemGroupDef[@Name=$parentDatasetName]">
            <xsl:variable name="datasetOID"
select="../odm:ItemGroupDef[@Name=$parentDatasetName]"/>
            <tr>
              <td colspan="8">
                <xsl:text>Related dataset: </xsl:text>
                <xsl:value-of
select="../odm:ItemGroupDef[@Name=$parentDatasetName]/odm:Description/odm:Translate
dText"/>
                <xsl:text> (</xsl:text>
                <a>
                  <xsl:attribute name="href">#IG.<xsl:value-of
select="$datasetOID/@OID"/></xsl:attribute>
                  <xsl:value-of select="$parentDatasetName"/>)
                </a>
              </td>
            </tr>
          </xsl:if>
        </xsl:if>

      </xsl:element>
    </div>
  </xsl:template>

        <!-- ********************************************************************* -->
        <!-- Template: AppendixValueList (handles the def:ValueListDef elements -->
        <!-- ********************************************************************* -->
        <xsl:template name="AppendixValueList">

          <xsl:if test="$g_seqValueListDefs">

            <xsl:call-template name="lineBreak"/>

            <a name="valuemeta"/>
                        <div class="containerbox" style="page-break-after:
always;">
                                    <xsl:element name="h1">
                                            <xsl:choose>
                                              <xsl:when
test="$g_ItemGroupDefPurpose='Analysis'">
                                                <xsl:text>Parameter Value
Lists</xsl:text>
                                              </xsl:when>
                                              <xsl:when
test="$g_ItemGroupDefPurpose='Tabulation'">
                                                <xsl:text>Value Level
```

```
Metadata</xsl:text>
                                        </xsl:when>
                                        <xsl:otherwise>
                                                <xsl:text>Value
Lists</xsl:text>
                                        </xsl:otherwise>
                                </xsl:choose>
                        </xsl:element>

                <xsl:for-each select="$g_seqValueListDefs">

                                <xsl:element name="div">
                                        <!-- page break after -->
                                        <xsl:attribute
name="class">containerbox</xsl:attribute>
                                        <xsl:attribute
name="id">VL.<xsl:value-of select="@OID"/></xsl:attribute>

                                        <xsl:variable
name="valueListDefOID" select="@OID"/>
                                        <xsl:variable name="valueListRef"
select="//odm:ItemDef/def:ValueListRef[@ValueListOID=$valueListDefOID]"/>
                                        <xsl:variable name="itemDefOID"
select="$valueListRef/../@OID"/>

                                        <xsl:element name="table">
                                                <xsl:attribute
name="summary">ValueList / ParameterList</xsl:attribute>

                                                <!-- set the legend (title)
-->
                                                <xsl:element
name="caption">
                                                        <xsl:choose>
                                                          <xsl:when
test="$g_ItemGroupDefPurpose='Analysis'"> Parameter Value List - </xsl:when>
                                                          <xsl:when
test="$g_ItemGroupDefPurpose='Tabulation'"> Value Level Metadata - </xsl:when>

<xsl:otherwise> Value List - </xsl:otherwise>
                                                        </xsl:choose>

                                                        <xsl:choose>
                                                                <xsl:when
test="//odm:ItemRef[@ItemOID=$itemDefOID]/../@Name">
                                                                  <!--
ValueList attached to an ItemGroup Item -->

<xsl:value-of select="//odm:ItemRef[@ItemOID=$itemDefOID]/../@Name"/>
                                                                </xsl:when>
```

```
<xsl:otherwise>
                                                                    <!--
ValueList attached to a ValueList Item -->

<xsl:value-of select="//odm:ItemRef[@ItemOID=$itemDefOID]/../@OID"/>
                  </xsl:otherwise>
                                                                        </xsl:choose>

                                                                  <xsl:text> [</xsl:text>
                                                                  <xsl:value-of
select="$valueListRef/../@Name"/>
                                                                  <xsl:text>]</xsl:text>

                                                                </xsl:element>

                                                                <tr class="header">

                                                                        <th
scope="col">Variable</th>
                                                                        <th
scope="col">Where</th>
                                                                        <th
scope="col">Type</th>
                                                                        <th scope="col"
abbr="Length" class="length">Length / Display Format</th>
                                                                        <th scope="col"
abbr="Format">Controlled Terms or Format</th>
                                                                  <xsl:if
test="$g_ItemGroupDefPurpose != 'Analysis'">
                                                                    <th
scope="col">Origin</th>
                                                                  </xsl:if>
                                                                  <xsl:choose>
                                                                    <xsl:when
test="$g_ItemGroupDefPurpose='Analysis'">
                                                                      <th
scope="col">Source/Derivation/Comment</th>
                                                                    </xsl:when>
                                                                    <xsl:when
test="$g_ItemGroupDefPurpose='Tabulation'">
                                                                      <th
scope="col">Derivation/Comment</th>
                                                                    </xsl:when>
                                                                    <xsl:otherwise>
                                                                      <th
scope="col">Derivation/Comment</th>
                                                                    </xsl:otherwise>
                                                                  </xsl:choose>
                                                                </tr>
```

```
                                                        <!-- Get the individual
data points -->
                                                        <xsl:for-each
select="./odm:ItemRef">
                                                                <xsl:variable
name="ItemRef" select="."/>
                                                                <xsl:variable
name="valueDefOid" select="@ItemOID"/>
                                                                <xsl:variable
name="valueDef" select="../../odm:ItemDef[@OID=$valueDefOid]"/>

                                                                <xsl:variable
name="vlOID" select="../@OID"/>
                                                                <xsl:variable
name="parentDef"
select="../../odm:ItemDef/def:ValueListRef[@ValueListOID=$vlOID]"/>
                                                                <xsl:variable
name="parentOID" select="$parentDef/../@OID"/>
                                                                <xsl:variable
name="ParentVName" select="$parentDef/../@Name"/>

                                                                <xsl:variable
name="ValueItemGroupOID"

select="$g_seqItemGroupDefs/odm:ItemRef[@ItemOID=$parentOID]/../@OID"/>

                                                                <xsl:variable
name="whereOID" select="./def:WhereClauseRef/@WhereClauseOID"/>
                                                                        <xsl:variable
name="whereDef" select="$g_seqWhereClauseDefs[@OID=$whereOID]"/>
                                                                <xsl:variable
name="whereRefItemOID" select="$whereDef/odm:RangeCheck/@def:ItemOID"/>
                                                                <xsl:variable
name="whereRefItem"

select="$g_seqItemDefs[@OID=$whereRefItemOID]/@Name"/>
                                                                <xsl:variable
name="whereOP" select="$whereDef/odm:RangeCheck/@Comparator"/>
                                                                <xsl:variable
name="whereVal" select="$whereDef/odm:RangeCheck/odm:CheckValue"/>

                                                                <tr>


<xsl:call-template name="rowClass">

<xsl:with-param name="rowNum" select="position()"/>

</xsl:call-template>
```

```
                                                                    <!-- first
column: Source Variable column -->
                                                                    <td>

<xsl:value-of select="$ParentVName"/>
                                                                    </td>
                                                                    <!-- second
column: 'WhereClause' column -->
                                                                    <td>

<xsl:call-template name="assembleWhereText">

<xsl:with-param name="ValueItemRef" select="$ItemRef"/>

<xsl:with-param name="ItemGroupLink" select="$ValueItemGroupOID"/>

<xsl:with-param name="decode" select="1"/>

<xsl:with-param name="break" select="1"/>

</xsl:call-template>


<!-- PPD (b) (4), (b) (6) 21Apr2017: Removed presentation of QVAL label to be consistent
w other VLM presentation -->


<!-- xsl:if test="$ParentVName='QVAL'"> (<xsl:value-of -->

<!--    select="$valueDef/odm:Description/odm:TranslatedText"/>) </xsl:if>      -->

                                                                    </td>

                                                                    <!-- Third
column: datatype -->
                                                                    <td
class="datatype">

<xsl:value-of select="$valueDef/@DataType"/>
                                                                    </td>

                                                                <!-- Fourth
column: Length/DisplayFormat -->
                                                                <td
class="number">

<xsl:choose>

<xsl:when test="$valueDef/@def:DisplayFormat">
```

```
        <xsl:value-of select="$valueDef/@def:DisplayFormat"/>
</xsl:when>
<xsl:otherwise>
        <xsl:value-of select="$valueDef/@Length"/>
</xsl:otherwise>
</xsl:choose>
                                                                    </td>
                                                                    <!-- Fifth
column: Controlled Terms or Format -->
                                                                    <td>
<xsl:call-template name="linkDecodeList">
<xsl:with-param name="itemDef" select="$valueDef"/>
</xsl:call-template>

<xsl:call-template name="displayISO8601">
<xsl:with-param name="itemDef" select="$valueDef"/>
</xsl:call-template>
                                                                    </td>

                                                                    <!--
********************************************** -->
                                                                    <!-- Origin
Column for ValueDefs (when not ADaM)         -->
                                                                    <!--
********************************************** -->
                                                                <xsl:if
test="$g_ItemGroupDefPurpose != 'Analysis'">
                                                                  <td>
<xsl:call-template name="displayItemDefOrigin">
<xsl:with-param name="itemDef" select="$valueDef"/>
</xsl:call-template>
                                                                  </td>
```

```
                                                                    </xsl:if>


                                                                          <!--
********************************************** -->
                                                                    <!--
Source/Derivation/Comment                                 -->
                                                                          <!--
********************************************** -->
                                                                          <td>

<xsl:variable name="whereclausecmntOID" select="$whereDef/@def:CommentOID"/>
                                                                            <xsl:if
test="$whereclausecmntOID">

<xsl:call-template name="displayItemDefComment">

<xsl:with-param name="itemDef" select="$whereDef"/>

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

                                                                            <xsl:if
test="$g_ItemGroupDefPurpose = 'Analysis'">

                                                                    <xsl:variable
name="Origin" select="$valueDef/def:Origin"/>
                                                                            <xsl:if
test="$Origin">

<xsl:choose>

<xsl:when test="$Origin[@Type='Predecessor']"> Predecessor: <xsl:value-of

select="$valueDef/def:Origin/odm:Description/odm:TranslatedText"/>

</xsl:when>

<xsl:otherwise>

<xsl:value-of select="$Origin/@Type"/>

<xsl:text>: </xsl:text>

</xsl:otherwise>

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

```
<xsl:variable name="methodOID" select="$ItemRef/@MethodOID"/>
                                                                    <xsl:if
test="$methodOID">
<xsl:call-template name="displayItemDefMethod">
<xsl:with-param name="itemRef" select="$ItemRef"/>
</xsl:call-template>
                                                                    </xsl:if>

                                                                  <xsl:variable
name="cmntOID" select="$valueDef/@def:CommentOID"/>
                                                                  <xsl:if
test="$cmntOID">
<xsl:call-template name="displayItemDefComment">
<xsl:with-param name="itemDef" select="$valueDef"/>
</xsl:call-template>
                                                                  </xsl:if>

                                                                        </td>
                                                                </tr>
                                                                <!-- end of loop
over all def:ValueListDef elements -->

                                                                <!--
**************************************************  -->
                                                                <!-- Link back to
the dataset from QNAM                    -->
                                                                <!-- For those
domains with Suplemental Qualifiers        -->
                                                                <!--
**************************************************  -->


                                                             </xsl:for-each>
                                                             <!-- end of loop over all
ValueListDefs -->
                                                    </xsl:element>
                                             </xsl:element>

                                    </xsl:for-each>
                            </div>

                            <xsl:call-template name="linkTop"/>
                    <xsl:call-template name="lineBreak"/>
```

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

  <!-- ************************************* -->
  <!-- Code List Items                       -->
  <!-- ************************************* -->
  <xsl:template name="AppendixCodeLists">

    <xsl:if test="$g_seqCodeLists[odm:CodeListItem|odm:EnumeratedItem]">

      <a id="decodelist"/>
      <div class="containerbox" style="page-break-after: always;">
        <h1>Controlled Terms</h1>

        <xsl:for-each
select="$g_seqCodeLists[odm:CodeListItem|odm:EnumeratedItem]">

          <xsl:choose>
            <xsl:when test="./odm:CodeListItem">
              <xsl:call-template name="displayCodeListItemsTable"/>
            </xsl:when>
            <xsl:when test="./odm:EnumeratedItem">
              <xsl:call-template name="displayEnumeratedItemsTable"/>
            </xsl:when>
            <xsl:otherwise />
          </xsl:choose>

       </xsl:for-each>

        <xsl:call-template name="linkTop"/>

      </div>
    </xsl:if>
  </xsl:template>

  <!-- ************************************* -->
  <!-- External Dictionaries                 -->
  <!-- ************************************* -->
  <xsl:template name="AppendixExternalCodeLists">

    <xsl:if test="$g_seqCodeLists[odm:ExternalCodeList]">

      <a id="externaldictionary"/>
      <h1 class="invisible">External Dictionaries</h1>
      <div class="containerbox" style="page-break-after: always;">

        <xsl:element name="table">
          <xsl:attribute name="summary">External Dictionaries (MedDra, WHODRUG,
...)</xsl:attribute>
```

```
          <caption class="header">External Dictionaries</caption>

          <tr class="header">
            <th scope="col">Reference Name</th>
            <th scope="col">External Dictionary</th>
            <th scope="col">Dictionary Version</th>
          </tr>

          <xsl:for-each select="$g_seqCodeLists/odm:ExternalCodeList">

            <xsl:element name="tr">

              <!-- Create an anchor -->
              <xsl:attribute name="id">CL.<xsl:value-of
select="../@OID"/></xsl:attribute>

              <xsl:call-template name="rowClass">
                <xsl:with-param name="rowNum" select="position()"/>
              </xsl:call-template>

              <td><xsl:value-of select="../@Name"/> (<xsl:value-of
select="../@OID"/>)</td>
              <td>
                <xsl:choose>
                  <xsl:when test="@href">
                    <a>
                      <xsl:attribute name="href"><xsl:value-of
select="@href"/></xsl:attribute>
                      <xsl:value-of select="@Dictionary"/>
                    </a>
                  </xsl:when>
                  <xsl:otherwise>
                    <xsl:value-of select="@Dictionary"/>
                  </xsl:otherwise>
                </xsl:choose>
              </td>
              <td><xsl:value-of select="@Version"/></td>

            </xsl:element>
          </xsl:for-each>
        </xsl:element>
      </div>
      <xsl:call-template name="linkTop"/>

    </xsl:if>
  </xsl:template>

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

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

    <xsl:if test="$g_seqMethodDefs">

      <a id="compmethod"/>
      <div class="containerbox" style="page-break-after: always;">
        <xsl:element name="h1">
          <xsl:attribute name="class">invisible</xsl:attribute>
          <xsl:choose>
            <xsl:when test="$g_ItemGroupDefPurpose='Tabulation'">
              <xsl:text>Computational Algorithms</xsl:text>
            </xsl:when>
            <xsl:when test="$g_ItemGroupDefPurpose='Analysis'">
              <xsl:text>Analysis Derivations</xsl:text>
            </xsl:when>
            <xsl:otherwise>Methods</xsl:otherwise>
          </xsl:choose>
        </xsl:element>

        <xsl:element name="table">
          <xsl:attribute name="summary">Computational Algorithms / Analysis
Derivations</xsl:attribute>

          <!-- set the legend (title) -->
          <xsl:element name="caption">
            <xsl:attribute name="class">header</xsl:attribute>
            <xsl:choose>
              <xsl:when test="$g_ItemGroupDefPurpose='Tabulation'">
                <xsl:text>Computational Algorithms</xsl:text>
              </xsl:when>
              <xsl:when test="$g_ItemGroupDefPurpose='Analysis'">
                <xsl:text>Analysis Derivations</xsl:text>
              </xsl:when>
              <xsl:otherwise>Methods</xsl:otherwise>
            </xsl:choose>
          </xsl:element>

          <tr class="header">
            <th scope="col">Method</th>
            <th scope="col">Type</th>
            <th scope="col">Description</th>
          </tr>
          <xsl:for-each select="$g_seqMethodDefs">

            <xsl:element name="tr">

              <!-- Create an anchor -->
              <xsl:attribute name="id">MT.<xsl:value-of
select="@OID"/></xsl:attribute>
```

```
              <xsl:call-template name="rowClass">
                <xsl:with-param name="rowNum" select="position()"/>
              </xsl:call-template>

              <td>
                <xsl:value-of select="@Name"/>
              </td>
              <td>
                <xsl:value-of select="@Type"/>
              </td>
              <td>
                <xsl:value-of select="normalize-space(.)"/>

                <xsl:for-each select="./def:DocumentRef">
                  <xsl:variable name="leafID" select="@leafID"/>
                  <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

                  <xsl:variable name="title" select="$leaf/def:title"/>
                  <xsl:variable name="href" select="$leaf/@xlink:href"/>
                  <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
                  <xsl:variable name="PageRefs"
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
                  <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
                  <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

                  <p class="linebreakcell">
                    <xsl:call-template name="createHyperLink">
                      <xsl:with-param name="href" select="$href"/>
                      <xsl:with-param name="PageRefType" select="$PageRefType"/>
                      <xsl:with-param name="PageRefs" select="$PageRefs"/>
                      <xsl:with-param name="PageFirst" select="$PageFirst"/>
                      <xsl:with-param name="PageLast" select="$PageLast"/>
                      <xsl:with-param name="title" select="$title"/>
                    </xsl:call-template>
                  </p>

                </xsl:for-each>

              </td>
            </xsl:element>
          </xsl:for-each>
        </xsl:element>
      </div>
      <xsl:call-template name="linkTop"/>

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

```
  <!-- ************************************************************ -->
  <!-- Comments                                                     -->
  <!-- ************************************************************ -->
  <xsl:template name="AppendixComments">

    <xsl:if test="$g_seqCommentDefs">

      <a id="comment"/>
      <div class="containerbox" style="page-break-after: always;">
        <h1 class="invisible">Comments</h1>

        <xsl:element name="table">
          <xsl:attribute name="summary">ItemGroup, ItemDef and WhereClauseDef
Comments</xsl:attribute>
          <caption class="header">Comments</caption>
          <!-- set the legend (title) -->

          <tr class="header">
            <th scope="col">CommentOID</th>
            <th scope="col">Description</th>
          </tr>
          <xsl:for-each select="$g_seqCommentDefs">
            <xsl:element name="tr">

              <!-- Create an anchor -->
              <xsl:attribute name="id">COMM.<xsl:value-of
select="@OID"/></xsl:attribute>

              <xsl:call-template name="rowClass">
                <xsl:with-param name="rowNum" select="position()"/>
              </xsl:call-template>

              <td>
                <xsl:value-of select="@OID"/>
              </td>
              <td>
                <xsl:value-of select="normalize-space(.)"/>

                <xsl:for-each select="./def:DocumentRef">
                  <xsl:variable name="leafID" select="@leafID"/>
                  <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

                  <xsl:variable name="title" select="$leaf/def:title"/>
                  <xsl:variable name="href" select="$leaf/@xlink:href"/>
                  <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
                  <xsl:variable name="PageRefs"
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
                  <xsl:variable name="PageFirst"
```

```
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
                  <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

                  <p class="linebreakcell">
                    <xsl:call-template name="createHyperLink">
                      <xsl:with-param name="href" select="$href"/>
                      <xsl:with-param name="PageRefType" select="$PageRefType"/>
                      <xsl:with-param name="PageRefs" select="$PageRefs"/>
                      <xsl:with-param name="PageFirst" select="$PageFirst"/>
                      <xsl:with-param name="PageLast" select="$PageLast"/>
                      <xsl:with-param name="title" select="$title"/>
                    </xsl:call-template>
                  </p>

                </xsl:for-each>
              </td>
            </xsl:element>
          </xsl:for-each>
        </xsl:element>
      </div>
      <xsl:call-template name="linkTop"/>

    </xsl:if>
  </xsl:template>

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

  <!-- Templates for special features like hyperlinks      -->
  <!--                                                     -->
  <!-- ************************************************* -->

  <!-- ******************************************************* -->
  <!-- Hypertext Link to CRF Pages (if necessary)              -->
  <!-- New mechanism: transform all numbers found in the string -->
  <!-- to hyperlinks                                           -->
  <!-- ******************************************************* -->
  <xsl:template name="crfPageNumbers2Hyperlinks">
    <xsl:param name="DefOriginString"/>
    <xsl:param name="Separator"/>
    <xsl:variable name="OriginString" select="$DefOriginString"/>

    <xsl:variable name="first">
      <xsl:choose>
        <xsl:when test="contains($OriginString,$Separator)">
          <xsl:value-of select="substring-before($OriginString,$Separator)"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="$OriginString"/>
        </xsl:otherwise>
```

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

  <xsl:variable name="rest" select="substring-after($OriginString,$Separator)"/>
  <xsl:variable name="stringlengthfirst" select="string-length($first)"/>

  <xsl:if test="string-length($first) > 0">
    <xsl:choose>
      <xsl:when test="number($first)">
        <!-- it is a number, create the hyperlink -->
        <xsl:call-template name="crfSinglePageHyperlink">
          <xsl:with-param name="pagenumber" select="$first"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <!-- it is not a number -->
        <xsl:value-of select="$first"/>
        <xsl:value-of select="$Separator"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>

  <!-- split up the second part in words (recursion) -->
  <xsl:if test="string-length($rest) > 0">

    <xsl:choose>
      <xsl:when test="contains($rest,$Separator)">
          <xsl:call-template name="crfPageNumbers2Hyperlinks">
          <xsl:with-param name="DefOriginString" select="$rest"/>
          <xsl:with-param name="Separator" select="' '"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$Separator"/>
        <xsl:text> </xsl:text>
        <xsl:call-template name="crfSinglePageHyperlink">
          <xsl:with-param name="pagenumber" select="$rest"/>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:template>

<!-- ******************************************************** -->
<!-- Hypertext Link to a single CRF Page                      -->
<!-- ******************************************************** -->
<xsl:template name="crfSinglePageHyperlink">
  <xsl:param name="pagenumber"/>
  <!-- create the hyperlink itself -->
  <xsl:if test="$g_MetaDataVersion/def:AnnotatedCRF">
```

```
    <xsl:for-each select="$g_MetaDataVersion/def:AnnotatedCRF/def:DocumentRef">
      <xsl:variable name="leafIDs" select="@leafID"/>
      <xsl:variable name="leaf" select="../../def:leaf[@ID=$leafIDs]"/>
      <a>
        <xsl:attribute name="href">
          <xsl:value-of select="concat($leaf/@xlink:href,'#page=',$pagenumber)"/>
        </xsl:attribute>
        <xsl:value-of select="$pagenumber"/>
      </a>
      <xsl:text> </xsl:text>
    </xsl:for-each>
  </xsl:if>
</xsl:template>

<!-- ******************************************************** -->
<!-- Hypertext Link to a CRF Page Named Destination           -->
<!-- ******************************************************** -->
<xsl:template name="crfNamedDestinationHyperlink">
  <xsl:param name="destination"/>
  <!-- create the hyperlink itself -->
  <xsl:if test="$g_MetaDataVersion/def:AnnotatedCRF">
    <xsl:for-each select="$g_MetaDataVersion/def:AnnotatedCRF/def:DocumentRef">
      <xsl:variable name="leafIDs" select="@leafID"/>
      <xsl:variable name="leaf" select="../../def:leaf[@ID=$leafIDs]"/>
      <a>
        <xsl:attribute name="href">
          <xsl:value-of select="concat($leaf/@xlink:href,'#',$destination)"/>
        </xsl:attribute>
        <xsl:value-of select="$destination"/>
      </a>
      <xsl:text> </xsl:text>
    </xsl:for-each>
  </xsl:if>
</xsl:template>

<!-- ******************************************************** -->
<!-- Hypertext Link to a single Page                          -->
<!-- ******************************************************** -->
<xsl:template name="linkSinglePageHyperlink">
  <xsl:param name="href"/>
  <xsl:param name="pagenumber"/>
  <xsl:param name="title"/>
  <!-- create the hyperlink itself -->
  <a class="external">
    <xsl:attribute name="href">
      <xsl:value-of select="concat($href,'#page=',$pagenumber)"/>
    </xsl:attribute>
    <xsl:value-of select="$title"/>
  </a>
  <xsl:text> </xsl:text>
```

```
</xsl:template>

<!-- ******************************************************** -->
<!-- Hypertext Link to a Named Destination                    -->
<!-- ******************************************************** -->
<xsl:template name="linkNamedDestinationHyperlink">
  <xsl:param name="href"/>
  <xsl:param name="destination"/>
  <xsl:param name="title"/>
  <!-- create the hyperlink itself -->
  <a class="external">
    <xsl:attribute name="href">
      <xsl:value-of select="concat($href,'#',$destination)"/>
    </xsl:attribute>
    <xsl:value-of select="$title"/>
  </a>
  <xsl:text> </xsl:text>
</xsl:template>


<!-- ******************************************************** -->
<!-- Hypertext Link to a Document                             -->
<!-- ******************************************************** -->
<xsl:template name="linkDocumentHyperlink">
  <xsl:param name="href"/>
  <xsl:param name="title"/>
  <!-- create the hyperlink itself -->
  <a class="external">
    <xsl:attribute name="href">
      <xsl:value-of select="$href"/>
    </xsl:attribute>
    <xsl:value-of select="$title"/>
  </a>
  <xsl:text> </xsl:text>
</xsl:template>

<!-- ******************************************************** -->
<!-- Hypertext Link to a Document                             -->
<!-- ******************************************************** -->
<xsl:template name="createHyperLink">
  <xsl:param name="href"/>
  <xsl:param name="PageRefType"/>
  <xsl:param name="PageRefs"/>
  <xsl:param name="PageFirst"/>
  <xsl:param name="PageLast"/>
  <xsl:param name="title"/>
  <xsl:choose>
    <xsl:when test="$PageRefType = $REFTYPE_PHYSICALPAGE">
      <xsl:call-template name="linkSinglePageHyperlink">
        <xsl:with-param name="href" select="$href"/>
```

```
          <xsl:with-param name="pagenumber" select="$PageRefs"/>
          <xsl:with-param name="title" select="$title"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:when test="$PageRefType = $REFTYPE_NAMEDDESTINATION">
        <xsl:call-template name="linkNamedDestinationHyperlink">
          <xsl:with-param name="href" select="$href"/>
          <xsl:with-param name="destination" select="$PageRefs"/>
          <xsl:with-param name="title" select="$title"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name="linkDocumentHyperlink">
          <xsl:with-param name="href" select="$href"/>
          <xsl:with-param name="title" select="$title"/>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>


  <!-- ************************************************************ -->
  <!-- ItemGroup Comment                                            -->
  <!-- ************************************************************ -->
  <xsl:template name="displayItemGroupComment">
    <xsl:if test="@def:CommentOID">
      <xsl:variable name="cmntOID" select="@def:CommentOID"/>
      <xsl:variable name="Comment" select="$g_seqCommentDefs[@OID=$cmntOID]"/>
      <xsl:variable name="ItemGroupComment">
        <xsl:value-of
select="$g_seqCommentDefs[@OID=$cmntOID]/odm:Description/odm:TranslatedText"/>
      </xsl:variable>
      <xsl:value-of select="$ItemGroupComment"/>
      <xsl:for-each select="$Comment/def:DocumentRef">
        <xsl:variable name="leafID" select="@leafID"/>
        <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

        <xsl:variable name="title" select="$leaf/def:title"/>
        <xsl:variable name="href" select="$leaf/@xlink:href"/>
        <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
        <xsl:variable name="PageRefs"
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
        <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
        <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

        <p class="linebreakcell">
          <xsl:call-template name="createHyperLink">
```

```
            <xsl:with-param name="href" select="$href"/>
            <xsl:with-param name="PageRefType" select="$PageRefType"/>
            <xsl:with-param name="PageRefs" select="$PageRefs"/>
            <xsl:with-param name="PageFirst" select="$PageFirst"/>
            <xsl:with-param name="PageLast" select="$PageLast"/>
            <xsl:with-param name="title" select="$title"/>
          </xsl:call-template>
        </p>

      </xsl:for-each>
    </xsl:if>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- ItemDef Origin                                             -->
  <!-- ********************************************************** -->
  <xsl:template name="displayItemDefOrigin">
    <xsl:param name="itemDef"/>

        <xsl:for-each select="$itemDef/def:Origin">

    <!-- translate the value of the def:Origin/@Type attribute to uppercase
              in order to see whether it contains the word "CRF" case-insensitive
-->
    <xsl:variable name="ORIGIN_UPPERCASE"
select="translate(@Type,$LOWERCASE,$UPPERCASE)"/>
    <xsl:choose>
      <!-- create a set of hyperlinks to CRF pages -->
      <!-- PPD (b) (4), (b) (6) 05May2016: Adjusted to look for Contains -->
      <xsl:when test="contains($ORIGIN_UPPERCASE, 'CRF')">

        <xsl:variable name="PageRefType"
select="normalize-space(def:DocumentRef/def:PDFPageRef/@Type)"/>
        <xsl:variable name="PageRefs"
select="normalize-space(def:DocumentRef/def:PDFPageRef/@PageRefs)"/>
        <xsl:variable name="PageFirst"
select="normalize-space(def:DocumentRef/def:PDFPageRef/@FirstPage)"/>
        <xsl:variable name="PageLast"
select="normalize-space(def:DocumentRef/def:PDFPageRef/@LastPage)"/>

        <xsl:value-of select="@Type"/>
        <xsl:choose>
          <xsl:when test="$PageRefType = $REFTYPE_NAMEDDESTINATION">
            <xsl:text> Page </xsl:text>
          </xsl:when>
          <xsl:otherwise>
            <xsl:choose>
             <xsl:when test="contains($PageRefs, ' ')">
                <xsl:text> Pages </xsl:text>
              </xsl:when>
```

```
              <xsl:when test="string-length($PageFirst) &gt; 0 and
string-length($PageLast) &gt; 0">
                <xsl:text> Pages </xsl:text>
              </xsl:when>
              <xsl:otherwise>
                <xsl:text> Page </xsl:text>
              </xsl:otherwise>
            </xsl:choose>
          </xsl:otherwise>
        </xsl:choose>

          <xsl:choose>
            <xsl:when test="$PageRefType = $REFTYPE_PHYSICALPAGE">
              <xsl:call-template name="crfPageNumbers2Hyperlinks">
                <xsl:with-param name="DefOriginString">
                  <xsl:choose>
                    <xsl:when test="$PageRefs"><xsl:value-of
select="normalize-space($PageRefs)"/>
                    </xsl:when>
                    <xsl:when test="$PageFirst"><xsl:value-of
select="normalize-space(concat($PageFirst, '-', $PageLast))"/>
                    </xsl:when>
                    <xsl:otherwise>
                    </xsl:otherwise>
                  </xsl:choose>
                </xsl:with-param>
                <xsl:with-param name="Separator">
                  <xsl:choose>
                    <xsl:when test="$PageRefs"><xsl:value-of select="' '"/>
                    </xsl:when>
                    <xsl:when test="$PageFirst"><xsl:value-of select="'-'"/>
                    </xsl:when>
                    <xsl:otherwise>
                    </xsl:otherwise>
                  </xsl:choose>
                </xsl:with-param>
             </xsl:call-template>
            </xsl:when>
            <xsl:when test="$PageRefType = $REFTYPE_NAMEDDESTINATION">
              <xsl:call-template name="crfNamedDestinationHyperlink">
                <xsl:with-param name="destination" select="$PageRefs"/>
              </xsl:call-template>
            </xsl:when>
            <xsl:otherwise/>
          </xsl:choose>

      </xsl:when>
      <!-- all other cases, just print the content from the 'Origin' attribute -->
      <xsl:otherwise>
        <xsl:value-of select="@Type"/>
```

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

                <xsl:if test="position() != last()">
                        <xsl:text>, </xsl:text>
                </xsl:if>

        </xsl:for-each>
  </xsl:template>

        <!-- ************************************** -->
        <!-- Display CodeList table                 -->
        <!-- ************************************** -->
        <xsl:template name="displayCodeListItemsTable">
                <xsl:variable name="n_extended"
select="count(odm:CodeListItem/@def:ExtendedValue)"/>

                <div class="codelist">
                        <xsl:attribute name="id">CL.<xsl:value-of
select="@OID"/></xsl:attribute>

                        <xsl:element name="table">
                                <xsl:attribute name="summary">Controlled Term -
<xsl:value-of select="@Name"/></xsl:attribute>

                                <caption>
                                        <xsl:value-of select="@Name"/>
                                        <xsl:text> [</xsl:text>
                                        <xsl:value-of select="@OID"/>
                                        <xsl:if test="./odm:Alias/@Context =
'nci:ExtCodeID'">
                                                <span class="nci">, <xsl:value-of
select="./odm:Alias/@Name"/></span>
                                        </xsl:if>
                                        <xsl:text>]</xsl:text>
                                </caption>

                                <tr class="header">
                                        <th scope="col"
class="codedvalue">Permitted Value (Code)</th>
                                        <th scope="col">Display Value (Decode)</th>
                                </tr>

                                <xsl:for-each select="./odm:CodeListItem">
                                        <xsl:sort data-type="number" select="@Rank"
order="ascending"/>
                                        <xsl:sort data-type="number"
select="@OrderNumber" order="ascending"/>
                                        <xsl:element name="tr">
```

```
                                                <xsl:call-template name="rowClass">
                                                        <xsl:with-param
name="rowNum" select="position()"/>
                                                </xsl:call-template>
                                                <td>
                                                        <xsl:value-of
select="@CodedValue"/>
                                                        <xsl:if
test="./odm:Alias/@Context = 'nci:ExtCodeID'">
                <xsl:text> [</xsl:text>
                <span class="nci"><xsl:value-of select="./odm:Alias/@Name"/></span>
                <xsl:text>]</xsl:text>
              </xsl:if>
                                                  <xsl:if
test="@def:ExtendedValue='Yes'">
                                                    <xsl:text> [</xsl:text>
                <span class="extended">*</span>
                <xsl:text>]</xsl:text>
              </xsl:if>
                                                </td>
                                                <td>
                                                        <xsl:value-of
select="./odm:Decode/odm:TranslatedText"/>
                                                </td>
                                        </xsl:element>
                                </xsl:for-each>
                        </xsl:element>
                        <xsl:if test="$n_extended &gt; 0">
                                <p class="footnote"><span class="super">*</span>
Extended Value</p>
                        </xsl:if>
                </div>
        </xsl:template>

        <!-- ************************************** -->
        <!-- Enumerated Items Table                 -->
        <!-- ************************************** -->
        <xsl:template name="displayEnumeratedItemsTable">
                <xsl:variable name="n_extended"
select="count(odm:EnumeratedItem/@def:ExtendedValue)"/>

                <div class="codelist">
                        <xsl:attribute name="id">CL.<xsl:value-of
select="@OID"/></xsl:attribute>

                        <xsl:element name="table">
                                <xsl:attribute name="summary">Code List -
<xsl:value-of select="@Name"/></xsl:attribute>

                                <caption>
```

```
                                        <xsl:value-of select="@Name"/>
                                        <xsl:text> [</xsl:text>
                                        <xsl:value-of select="@OID"/>
                                        <xsl:if test="./odm:Alias/@Context =
'nci:ExtCodeID'">
                                                <span class="nci">, <xsl:value-of
select="./odm:Alias/@Name"/></span>
                                        </xsl:if>
                                        <xsl:text>]</xsl:text>
                                </caption>

                                <tr class="header">
                                        <th scope="col">Permitted Value (Code)</th>
                                </tr>

                                <xsl:for-each select="./odm:EnumeratedItem">
                                        <xsl:sort data-type="number" select="@Rank"
order="ascending"/>
                                        <xsl:sort data-type="number"
select="@OrderNumber" order="ascending"/>

                                        <xsl:element name="tr">
                                                <xsl:call-template name="rowClass">
                                                        <xsl:with-param
name="rowNum" select="position()"/>
                                                </xsl:call-template>
                                                <td>
                                                        <xsl:value-of
select="@CodedValue"/>
                                                  <xsl:if
test="./odm:Alias/@Context = 'nci:ExtCodeID'">
                                                    <xsl:text> [</xsl:text>
                                                    <span class="nci"><xsl:value-of
select="./odm:Alias/@Name"/></span>
                                                    <xsl:text>]</xsl:text>
                                                  </xsl:if>
                                                  <xsl:if
test="@def:ExtendedValue='Yes'">
                                                    <xsl:text> [</xsl:text>
                                                    <span class="extended">*</span>
                                                    <xsl:text>]</xsl:text>
                                                  </xsl:if>
                                                </td>
                                        </xsl:element>
                                </xsl:for-each>
                        </xsl:element>
                        <xsl:if test="$n_extended &gt; 0">
                                <p class="footnote"><span class="super">*</span>
Extended Value</p>
                        </xsl:if>
```

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

        <!-- ********************************************************** -->
  <!-- ItemDef Comment                                                -->
  <!-- ********************************************************** -->
  <xsl:template name="displayItemDefComment">

    <xsl:param name="itemDef"/>

    <xsl:if test="$itemDef/@def:CommentOID">
      <xsl:variable name="cmntOID" select="$itemDef/@def:CommentOID"/>
      <xsl:variable name="Comment" select="$g_seqCommentDefs[@OID=$cmntOID]"/>
      <xsl:variable name="ItemDefComment">
        <xsl:value-of
select="$g_seqCommentDefs[@OID=$cmntOID]/odm:Description/odm:TranslatedText"/>
      </xsl:variable>
      <p class="linebreakcell"><xsl:value-of select="$ItemDefComment"/></p>

      <xsl:for-each select="$Comment/def:DocumentRef">
        <xsl:variable name="leafID" select="@leafID"/>
        <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

        <xsl:variable name="title" select="$leaf/def:title"/>
        <xsl:variable name="href" select="$leaf/@xlink:href"/>
        <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
        <xsl:variable name="PageRefs"
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
        <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
        <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

        <p class="linebreakcell">
          <xsl:call-template name="createHyperLink">
            <xsl:with-param name="href" select="$href"/>
            <xsl:with-param name="PageRefType" select="$PageRefType"/>
            <xsl:with-param name="PageRefs" select="$PageRefs"/>
            <xsl:with-param name="PageFirst" select="$PageFirst"/>
            <xsl:with-param name="PageLast" select="$PageLast"/>
            <xsl:with-param name="title" select="$title"/>
          </xsl:call-template>
        </p>

      </xsl:for-each>
    </xsl:if>
  </xsl:template>

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

```
  <!-- ItemDef Method                                                    -->
  <!-- ************************************************************** -->
  <xsl:template name="displayItemDefMethod">
    <xsl:param name="itemRef"/>
    <!-- if there is a reference to a ComputationalMethod, add it to the Comments
column -->
    <xsl:if test="$itemRef/@MethodOID">
      <xsl:variable name="MethodOID" select="$itemRef/@MethodOID"/>
      <xsl:variable name="Method" select="$g_seqMethodDefs[@OID=$MethodOID]"/>
      <xsl:variable name="MethodComment">
        <xsl:value-of select="$Method/odm:Description/odm:TranslatedText"/>
      </xsl:variable>

      <div class="linebreakcell">
        <xsl:value-of select="$MethodComment"/>
      </div>

      <xsl:for-each select="$Method/def:DocumentRef">
        <xsl:variable name="leafID" select="$Method/def:DocumentRef/@leafID"/>
        <xsl:variable name="leaf" select="$g_seqleafs[@ID=$leafID]"/>

        <xsl:variable name="title" select="$leaf/def:title"/>
        <xsl:variable name="href" select="$leaf/@xlink:href"/>
        <xsl:variable name="PageRefType"
select="normalize-space(def:PDFPageRef/@Type)"/>
        <xsl:variable name="PageRefs"
select="normalize-space(def:PDFPageRef/@PageRefs)"/>
        <xsl:variable name="PageFirst"
select="normalize-space(def:PDFPageRef/@FirstPage)"/>
        <xsl:variable name="PageLast"
select="normalize-space(def:PDFPageRef/@LastPage)"/>

        <p class="linebreakcell">
          <xsl:call-template name="createHyperLink">
            <xsl:with-param name="href" select="$href"/>
            <xsl:with-param name="PageRefType" select="$PageRefType"/>
            <xsl:with-param name="PageRefs" select="$PageRefs"/>
            <xsl:with-param name="PageFirst" select="$PageFirst"/>
            <xsl:with-param name="PageLast" select="$PageLast"/>
            <xsl:with-param name="title" select="$title"/>
          </xsl:call-template>
        </p>

      </xsl:for-each>
    </xsl:if>
  </xsl:template>

  <!-- ************************************************************** -->
  <!-- Display Keys                                                   -->
  <!-- ************************************************************** -->
```

```
  <xsl:template name="displayKeys">
    <xsl:variable name="KeySequence" select="odm:ItemRef/@KeySequence"/>
    <xsl:variable name="n_keys" select="count($KeySequence)"/>
    <xsl:for-each select="odm:ItemRef">
      <xsl:sort select="@KeySequence" data-type="number" order="ascending"/>
      <xsl:if test="@KeySequence[ .!='' ]">
        <xsl:variable name="ItemOID" select="@ItemOID"/>
        <xsl:variable name="Name" select="$g_seqItemDefs[@OID=$ItemOID]"/>
        <xsl:value-of select="$Name/@Name"/>
        <xsl:if test="@KeySequence &lt; $n_keys">, </xsl:if>
      </xsl:if>
    </xsl:for-each>
  </xsl:template>

  <!-- ************************************************************ -->
  <!-- Link to Dataset                                              -->
  <!-- ************************************************************ -->
  <xsl:template name="linkDataset">
    <xsl:choose>
      <xsl:when test="@SASDatasetName">
        <xsl:value-of select="concat(./odm:Description/odm:TranslatedText, ' (',
@SASDatasetName, ') ')"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="concat(./odm:Description/odm:TranslatedText, ' (',
@Name, ') ')"/>
      </xsl:otherwise>
    </xsl:choose>

    <span class="dataset"><xsl:text>[Location: </xsl:text>
    <a>
      <xsl:attribute name="href">
        <xsl:value-of select="def:leaf/@xlink:href"/>
      </xsl:attribute>
      <xsl:value-of select="def:leaf/def:title"/>
    </a>
      <xsl:text>]</xsl:text></span>
  </xsl:template>

  <!-- ************************************************************ -->
  <!-- Where Text                                                   -->
  <!-- ************************************************************ -->
  <xsl:template name="assembleWhereText">
    <xsl:param name="ValueItemRef"/>
    <xsl:param name="ItemGroupLink"/>
    <xsl:param name="decode"/>
    <xsl:param name="break"/>

    <xsl:variable name="ValueRef" select="$ValueItemRef"/>
    <xsl:for-each select="$ValueRef/def:WhereClauseRef">
```

```
      <xsl:variable name="whereOID" select="./@WhereClauseOID"/>
      <xsl:variable name="whereDef"
select="$g_seqWhereClauseDefs[@OID=$whereOID]"/>
      <xsl:for-each select="$whereDef/odm:RangeCheck">

        <xsl:variable name="whereRefItemOID" select="./@def:ItemOID"/>
        <xsl:variable name="whereRefItemName"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@Name"/>
        <xsl:variable name="whereOP" select="./@Comparator"/>
        <xsl:variable name="whereRefItemCodeListOID"

select="$g_seqItemDefs[@OID=$whereRefItemOID]/odm:CodeListRef/@CodeListOID"/>
        <xsl:variable name="whereRefItemCodeList"
          select="$g_seqCodeLists[@OID=$whereRefItemCodeListOID]"/>

        <xsl:call-template name="linkItemGroupItem">
          <xsl:with-param name="ItemGroupOID" select="$ItemGroupLink"/>
          <xsl:with-param name="ItemOID" select="$whereRefItemOID"/>
          <xsl:with-param name="ItemName" select="$whereRefItemName"/>
        </xsl:call-template>

        <xsl:choose>
          <xsl:when test="$whereOP = 'IN' or $whereOP = 'NOTIN'">
            <xsl:text> </xsl:text>
            <xsl:variable name="Nvalues" select="count(./odm:CheckValue)"/>
            <xsl:choose>
              <xsl:when test="$whereOP='IN'">
                <xsl:text>IN</xsl:text>
              </xsl:when>
              <xsl:otherwise>
                <xsl:text>NOT IN</xsl:text>
              </xsl:otherwise>
            </xsl:choose>
            <xsl:text> (</xsl:text>
            <xsl:if test="$decode='1'"><br /></xsl:if>
            <xsl:for-each select="./odm:CheckValue">
              <xsl:variable name="CheckValueINNOTIN" select="."/>
              <span class="linebreakcell">
                <xsl:call-template name="displayValue">
                  <xsl:with-param name="Value" select="$CheckValueINNOTIN"/>
                  <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
                  <xsl:with-param name="decode" select="$decode"/>
                  <xsl:with-param name="CodeList" select="$whereRefItemCodeList"/>
                </xsl:call-template>
                <xsl:if test="position() != $Nvalues">
                  <xsl:value-of select="', '"/>
                </xsl:if>
              </span>
```

```
              <xsl:if test="$decode='1'"><br /></xsl:if>
            </xsl:for-each><xsl:text>) </xsl:text>
          </xsl:when>

          <xsl:when test="$whereOP = 'EQ'">
            <xsl:variable name="CheckValueEQ" select="./odm:CheckValue"/>
            <xsl:text> = </xsl:text>
            <xsl:call-template name="displayValue">
              <xsl:with-param name="Value" select="$CheckValueEQ"/>
              <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
              <xsl:with-param name="decode" select="$decode"/>
              <xsl:with-param name="CodeList" select="$whereRefItemCodeList"/>
            </xsl:call-template>
          </xsl:when>

          <xsl:when test="$whereOP = 'NE'">
            <xsl:variable name="CheckValueNE" select="./odm:CheckValue"/>
            <xsl:text> &#x2260; </xsl:text>
            <xsl:call-template name="displayValue">
              <xsl:with-param name="Value" select="$CheckValueNE"/>
              <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
              <xsl:with-param name="decode" select="$decode"/>
              <xsl:with-param name="CodeList" select="$whereRefItemCodeList"/>
            </xsl:call-template>
          </xsl:when>

          <xsl:otherwise>
            <xsl:variable name="CheckValueOTH" select="./odm:CheckValue"/>
            <xsl:text> </xsl:text>
            <xsl:choose>
              <xsl:when test="$whereOP='LT'">
                <xsl:text> &lt; </xsl:text>
              </xsl:when>
              <xsl:when test="$whereOP='LE'">
                <xsl:text> &lt;= </xsl:text>
              </xsl:when>
              <xsl:when test="$whereOP='GT'">
                <xsl:text> &gt; </xsl:text>
              </xsl:when>
              <xsl:when test="$whereOP='GE'">
                <xsl:text> &gt;= </xsl:text>
              </xsl:when>
              <xsl:otherwise>
                <xsl:value-of select="$whereOP"/>
              </xsl:otherwise>
            </xsl:choose>
            <xsl:call-template name="displayValue">
              <xsl:with-param name="Value" select="$CheckValueOTH"/>
```

```
              <xsl:with-param name="DataType"
select="$g_seqItemDefs[@OID=$whereRefItemOID]/@DataType"/>
              <xsl:with-param name="decode" select="$decode"/>
              <xsl:with-param name="CodeList" select="$whereRefItemCodeList"/>
            </xsl:call-template>
          </xsl:otherwise>
        </xsl:choose>

        <xsl:if test="$break='1'"><br/></xsl:if>
        <xsl:if test="position() != last()">
          <xsl:text> and </xsl:text>
        </xsl:if>

      </xsl:for-each>

      <xsl:if test="position() != last()">
        <xsl:text> or </xsl:text>
        <!-- only if this is not the last WhereRef in the ItemREf  -->
      </xsl:if>

    </xsl:for-each>
  </xsl:template>


  <!-- ************************************************************ -->
  <!-- displayValue                                                 -->
  <!-- ************************************************************ -->
  <xsl:template name="displayValue">
    <xsl:param name="Value"/>
    <xsl:param name="DataType"/>
    <xsl:param name="decode"/>
    <xsl:param name="CodeList"/>

    <xsl:if test="$DataType != 'integer' and $DataType != 'float'">
      <xsl:text>"</xsl:text><xsl:value-of select="$Value"/><xsl:text>"</xsl:text>
    </xsl:if>
    <xsl:if test="$DataType = 'integer' or $DataType = 'float'">
      <xsl:value-of select="$Value"/>
    </xsl:if>
    <xsl:if test="$decode='1'">
      <xsl:if test="$CodeList/odm:CodeListItem[@CodedValue=$Value]">
        <xsl:text> (</xsl:text>
        <xsl:value-of

select="$CodeList/odm:CodeListItem[@CodedValue=$Value]/odm:Decode/odm:TranslatedTex
t"/>
        <xsl:text>) </xsl:text>
      </xsl:if>
    </xsl:if>
  </xsl:template>
```

```
  <!-- ********************************************************** -->
  <!-- Where Text Parameter                                       -->
  <!-- ********************************************************** -->


  <!-- ********************************************************** -->
  <!-- Link to ItemGroup Item                                     -->
  <!-- ********************************************************** -->
  <xsl:template name="linkItemGroupItem">
    <xsl:param name="ItemGroupOID"/>
    <xsl:param name="ItemOID"/>
    <xsl:param name="ItemName"/>
    <xsl:choose>
      <xsl:when
test="$g_seqItemGroupDefs[@OID=$ItemGroupOID]/odm:ItemRef[@ItemOID=$ItemOID]">
        <xsl:variable name="ItemDescription"
select="$g_seqItemDefs[@OID=$ItemOID]/odm:Description/odm:TranslatedText"/>
        <a>
          <xsl:attribute name="href">#<xsl:value-of
select="$ItemGroupOID"/>.<xsl:value-of select="$ItemOID"/></xsl:attribute>
          <xsl:attribute name="title"><xsl:value-of
select="$ItemDescription"/></xsl:attribute>
          <xsl:value-of select="$ItemName"/>
        </a>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$ItemName"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- Link to DecodeList                                         -->
  <!-- ********************************************************** -->
  <xsl:template name="linkDecodeList">
    <xsl:param name="itemDef"/>
    <xsl:variable name="CODE" select="$itemDef/odm:CodeListRef/@CodeListOID"/>
    <xsl:variable name="CodeListDef" select="$g_seqCodeLists[@OID=$CODE]"/>
    <xsl:variable name="n_items"
select="count($CodeListDef/odm:CodeListItem|$CodeListDef/odm:EnumeratedItem)"/>
        <xsl:variable name="CodeListDataType" select="$CodeListDef/@DataType" />

    <xsl:if test="$itemDef/odm:CodeListRef">

      <xsl:choose>
        <xsl:when test="$n_items &lt; 0 and $CodeListDef/odm:CodeListItem">
                <xsl:text>[</xsl:text>

          <xsl:for-each select="$CodeListDef/odm:CodeListItem">
```

```
                <xsl:if test="$CodeListDataType='text'">
                        <xsl:value-of select="concat('&quot;', @CodedValue,
'&quot;')"/>
                </xsl:if>
                <xsl:if test="$CodeListDataType != 'text'">
                        <xsl:value-of select="@CodedValue"/>
                </xsl:if>
                <xsl:text> = </xsl:text>
            <xsl:value-of select="concat('&quot;', odm:Decode/odm:TranslatedText,
'&quot;')"/>
            <xsl:if test="@CodedValue !=
$CodeListDef/odm:CodeListItem[last()]/@CodedValue">, </xsl:if>
          </xsl:for-each>
          <xsl:text>]</xsl:text>
          <p class="linebreakcell"><xsl:text> &lt;</xsl:text>
            <a href="#CL.{$CodeListDef/@OID}">
              <xsl:value-of select="$CodeListDef/@Name"/>
            </a>&gt;</p>
        </xsl:when>
        <xsl:when test="$n_items &lt; 0 and $CodeListDef/odm:EnumeratedItem">
          <xsl:text>[</xsl:text>
          <xsl:for-each select="$CodeListDef/odm:EnumeratedItem">
                <xsl:if test="$CodeListDataType='text'">
                        <xsl:value-of select="concat('&quot;', @CodedValue,
'&quot;')"/>
                </xsl:if>
                <xsl:if test="$CodeListDataType != 'text'">
                        <xsl:value-of select="@CodedValue"/>
                </xsl:if>
                <xsl:if test="@CodedValue !=
$CodeListDef/odm:EnumeratedItem[last()]/@CodedValue">, </xsl:if>
          </xsl:for-each>
          <xsl:text>]</xsl:text>
          <p class="linebreakcell"><xsl:text> &lt;</xsl:text>
            <a href="#CL.{$CodeListDef/@OID}">
              <xsl:value-of select="$CodeListDef/@Name"/>
            </a>&gt;</p>
        </xsl:when>
        <xsl:otherwise>
            <xsl:choose>
              <xsl:when test="$g_seqCodeLists[@OID=$CODE]">
                <a href="#CL.{$CodeListDef/@OID}">
                  <xsl:value-of select="$CodeListDef/@Name"/>
                </a>
              </xsl:when>
              <xsl:otherwise>
                <xsl:value-of select="$itemDef/odm:CodeListRef/@CodeListOID"/>
              </xsl:otherwise>
            </xsl:choose>
        </xsl:otherwise>
```

```
      </xsl:choose>

    </xsl:if>
  </xsl:template>

        <!-- ************************************** -->
        <!-- Display ISO8601                        -->
        <!-- ************************************** -->
        <xsl:template name="displayISO8601">
                <xsl:param name="itemDef"/>
                <!-- when the datatype is 'date', 'time' or 'datetime'
                   or it is a -DUR (duration) variable, print 'ISO8601' in this
column -->
                <xsl:if
                        test="$itemDef/@DataType='date' or
                              $itemDef/@DataType='time' or
                              $itemDef/@DataType='datetime' or
                              $itemDef/@DataType='partialDate' or
                              $itemDef/@DataType='partialTime' or
                              $itemDef/@DataType='partialDatetime' or
                              $itemDef/@DataType='incompleteDatetime' or
                              $itemDef/@DataType='durationDatetime' or
                              ((string-length($itemDef/@Name) &gt; 1) and
$itemDef/@DataType='text' and

substring($itemDef/@Name,string-length($itemDef/@Name)-2,string-length($itemDef/@Na
me)) = 'DUR')">
                        <xsl:text>ISO8601</xsl:text>
                </xsl:if>
        </xsl:template>

        <!-- ********************************************************** -->
  <!-- Template:    rowClass                                      -->
  <!-- Description: This template sets the table row class attribute -->
  <!--              based on the specified table row number       -->
  <!-- ********************************************************** -->
  <xsl:template name="rowClass">
    <!-- rowNum: current table row number (1-based) -->
    <xsl:param name="rowNum"/>

    <!-- set the class attribute to "tableroweven" for even rows, "tablerowodd" for
odd rows -->
    <xsl:attribute name="class">
      <xsl:choose>
        <xsl:when test="$rowNum mod 2 = 0">
          <xsl:text>tableroweven</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>tablerowodd</xsl:text>
        </xsl:otherwise>
```

```
      </xsl:choose>
    </xsl:attribute>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- Template:    lineBreak                                     -->
  <!-- Description: This template adds a line break element      -->
  <!-- ********************************************************** -->
  <xsl:template name="lineBreak">
    <xsl:element name="br">
      <xsl:call-template name="noBreakSpace"/>
    </xsl:element>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- Template:    noBreakSpace                                  -->
  <!-- Description: This template returns a no-break-space character -->
  <!-- ********************************************************** -->
  <xsl:template name="noBreakSpace">
    <!-- equivalent to   -->
    <xsl:text/>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- Link to Top                                                -->
  <!-- ********************************************************** -->
  <xsl:template name="linkTop">
    <p class="linktop">Go to the <a href="#main">top</a> of the define.xml</p>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- Document Generation Date                                   -->
  <!-- ********************************************************** -->
  <xsl:template name="DocGenerationDate">
    <p class="documentinfo">Date of Define-XML document generation: <xsl:value-of
select="/odm:ODM/@CreationDateTime"/></p>
 </xsl:template>

  <!-- ********************************************************** -->
  <!-- StyleSheet Date                                            -->
  <!-- ********************************************************** -->
  <xsl:template name="StylesheetDate">
    <span class="stylesheetinfo">Stylesheet version: <xsl:value-of
select="$g_stylesheetVersion"/></span>
    <xsl:call-template name="lineBreak"/>
  </xsl:template>

  <!-- ********************************************************** -->
  <!-- Generate JavaScript                                        -->
  <!-- ********************************************************** -->
```

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

<script type="text/javascript">
<xsl:text disable-output-escaping="yes">
<![CDATA[
<!--
/**
 * With one argument, return the textContent or innerText of the element.
 * With two arguments, set the textContent or innerText of element to value.
 */
function textContent(element, value) {
  "use strict";
  var rtn;
  var content = element.textContent;  // Check if textContent is defined
  if (value === undefined) { // No value passed, so return current text
    if (content !== undefined) {
      rtn = content;
    } else {
      rtn = element.innerText;
    }
    return rtn;
  }
  else { // A value was passed, so set text
    if (content !== undefined) {
      element.textContent = value;
    } else {
      element.innerText = value;
    }
  }
}

var ITEM  = '\u00A0';
var OPEN  = '\u25BC';
var CLOSE = '\u25BA';

function toggle_submenu(e) {
  "use strict";
  if (textContent(e)===OPEN) {
    textContent(e, CLOSE);
  }
  else {
    textContent(e, OPEN);
  }

  var i;
  for (i=0; i < e.parentNode.childNodes.length; i++) {
    var c;
    c=e.parentNode.childNodes[i];
    if (c.tagName==='UL') {c.style.display=(c.style.display==='none') ? 'block' :
'none';}
```

```
   }
}

function reset_menus() {
"use strict";
  var li;
  var c;
  var i;
  var j;
  var li_tags = document.getElementsByTagName('LI');
  for (i=0; i < li_tags.length; i++) {
    li=li_tags[i];
    if ( li.className.match('hmenu-item') ){
      for (j=0; j < li.childNodes.length; j++) {
        c=li.childNodes[j];
        if ( c.tagName === 'SPAN' && c.className.match('hmenu-bullet') )
{textContent(c, ITEM);}
        }
      }
    if ( li.className.match('hmenu-submenu') ) {
      for (j=0; j < li.childNodes.length; j++) {
        c=li.childNodes[j];
        if ( c.tagName === 'SPAN' && c.className.match('hmenu-bullet') )
{textContent(c, CLOSE);}
        else if ( c.tagName === 'UL' ) { c.style.display = 'none'; }
        }
    }
  }
}
//-->
]]>
</xsl:text>
</script>
</xsl:template>


<!-- ************************************************************ -->
<!-- Generate CSS                                                 -->
<!-- ************************************************************ -->
<xsl:template name="GenerateCSS">
<style type="text/css">
  body{
  background-color:#FFFFFF;
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:62.5%;
  margin:0;
  padding:30px;
  }

  h1{
```

```
font-size:1.6em;
margin-left:0;
font-weight:bolder;
text-align:left;
color:#800000;
}

ul{
margin-left:0px;
}

a{
color:#0000FF;
text-decoration:underline;
}
a.visited{
color:#551A8B;
text-decoration:underline;
}
a:hover{
color:#FF9900;
text-decoration:underline;
}
a.tocItem{
color:#004A95;
text-decoration:none;
margin-top:2px;
font-size:1.4em;
}
a.tocItem.level2{
margin-left:15px;
}

#menu{
position:fixed;
left:0px;
top:10px;
width:20%;
height:96%;
bottom:0px;
overflow:auto;
background-color:#FFFFFF;
color:#000000;
border:0px none black;
text-align:left;
white-space:nowrap;
}

.hmenu li{
list-style:none;
```

```
line-height:200%;
padding-left:0;
}
.hmenu ul{
padding-left:14px;
margin-left:0;
}
.hmenu-item{
}
.hmenu-submenu{
}
.hmenu-bullet{
float:left;
width:16px;
color:#AAAAAA;
font-size:1.2em;
}

#main{
position:absolute;
left:22%;
top:0px;
overflow:auto;
color:#000000;
background-color:#FFFFFF;
}

#main .docinfo{
width:95%;
text-align:right;
padding: 0px 5px;
}

div.containerbox{
padding:0px;
margin:10px auto;
border:0px solid #999;
page-break-after:always;
}

div.codelist{
page-break-after:avoid;
}

table{
width:95%;
border-spacing:4px;
border:1px solid #000000;
background-color:#EEEEEE;
margin-top:5px;
```

```
border-collapse:collapse;
padding:5px;
empty-cells:show;
}

table caption{
border:0px solid #999999;
left:20px;
font-size:1.4em;
font-weight:bolder;
color:#800000;
margin:10px auto;
text-align:left;
}

table caption .dataset{
font-weight:normal;
}

table caption.header{
font-size:1.6em;
margin-left:0;
font-weight:bolder;
text-align:left;
color:#800000;
}

table tr{
border:1px solid #000000;
}

table tr.header{
background-color:#6699CC;
color:#FFFFFF;
font-weight:bold;
}

table th{
font-weight:bold;
vertical-align:top;
text-align:left;
padding:5px;
border:1px solid #000000;
font-size:1.3em;
}

table td{
vertical-align:top;
padding:5px;
border:1px solid #000000;
```

```
font-size:1.2em;
line-height:150%;
}

table th.codedvalue{
width:20%;
}
table th.length{
width:7%;
}
table td.datatype{
text-align:center;
}
table td.number{
text-align:right;
}
.tablerowodd{
background-color:#FFFFFF;
}
.tableroweven{
background-color:#E2E2E2;
}

.linebreakcell{
vertical-align:top;
margin-top:3px;
margin-bottom:3px;
white-space:pre;  /* CSS 2.0 */
      white-space: pre-wrap; /* CSS 2.1 */
      <!--white-space: pre-line; -->/* CSS 3.0 */
      white-space: -pre-wrap; /* Opera 4-6 */
      white-space: -o-pre-wrap; /* Opera 7 */
      white-space: -moz-pre-wrap; /* Mozilla */
      white-space: -hp-pre-wrap; /* HP Printers */
      word-wrap: break-word; /* IE 5+ */
}

.nci, .extended{
font-style:italic;
}
.super{
vertical-align:super;
}
.footnote{
font-size:1.2em;
}

.standard{
font-size:1.6em;
font-weight:bold;
```

```
text-align:left;
padding:15px;
margin-left:20px;
margin-top:40px;
margin-right:20px;
margin-bottom:20px;
color:#800000;
border:0px;
}

.study{
font-size:1.6em;
font-weight:bold;
text-align:left;
padding:0px;
margin-left:0px;
margin-top:00px;
margin-right:0px;
margin-bottom:0px;
color:#800000;
border:0px none;
}

.linktop{
font-size:1.2em;
margin-top:5px;
}
.documentinfo, .stylesheetinfo{
font-size:1.2em;
}

.invisible{
display:none;
}

span.error{
width:95%;
font-size:1.6em;
font-weight: bold;
padding:5px;
color:#FF0000;
border-spacing:4px;
border:2px solid #FF0000;
}
td.error{
color:#FF0000;
}

.arm-table{ background-color:#ecececc;}
```

```
  table th.label{
  width:13%;
  }
  .arm{
  margin-top:5px;
  margin-bottom:5px;
  margin-left:5px;
  margin-right:0;
  background-color:#C0C0C0;
  width:97%;
  }

  .title{ margin-left:5pt; }

  p.summaryresult{ margin-left:15px; margin-top:5px; margin-bottom:5px;}
  p.parameter{ margin-top:5px; margin-bottom:5px;}
  p.analysisvariable{ margin-top:5px; margin-bottom:5px;}
  .datareference{ margin-top:5px; margin-bottom:5px;}
  tr.analysisresult{ background-color:#6699CC; color:#FFFFFF; font-weight:bold;
border:1px solid black;}

  .code-context{
  padding:5px 0px;
  }
  .coderef{
  font-size:1.2em;
  line-height:150%;
  padding:5px;
  }
  .code{
  font-family:"Courier New", monospace, serif;
  font-size:1.2em;
  line-height:150%;
  white-space:pre;
  display:block;
  vertical-align:top;
  padding:5px;
  }


  dl.multiple-table
  {
  width:95%;
  padding: 5px 0px;
  font-size:0.8em;
  color:#000000;
  }

  dl.multiple-table dt
  {
```

```
clear: left;
float: left;
width: 200px;
margin: 0;
padding: 5px 5px 5px 0px;
font-weight: bold;
}

dl.multiple-table dd
{
margin-left: 210px;
padding: 5px;
font-weight: normal;
}

@media print{

body, h1, table caption, table caption.header{
color:#000000;
}

a:link,
a:visited{
background:transparent;
text-decoration:none;
color:#000000;
}
a.external:link:after,
#main a:visited:after{
content:" &lt;" attr(href) "&gt; ";
font-size:90%;
text-decoration:none;
font-weight:bold;
color:#808080;
}

table{
border-width:2px;
}

#menu,
.linktop, .stylesheetinfo{
display:none !important;
width:0px;
}
#main{
left:0px;
}

}
```

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

</xsl:stylesheet>
```