



Guide to XSLs for DED

DED XSL Guide

Version 2019-MAR

December 1, 2022

Distribution Notice:

This document has been approved for Public Release and is available for use without restriction.

Table of Contents

Chapter 1 - Introduction	1
1.1 - Purpose	1
Chapter 2 - XSL Files	2
2.1 - Ded-to-Docbook.xsl	2
2.2 - Xsd-to-Ded.xsl	7

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for DED. This document provides XSL files developed for DED.

Chapter 2 - XSL Files

2.1 - Ded-to-Docbook.xsl

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns:xd="http://www.oxygenxml.com/ns/doc/xsl"
                xmlns:ded="urn:us:gov:ic:ded"
                xmlns:edh="urn:us:gov:ic:edh"
                xmlns:arh="urn:us:gov:ic:arh"
                exclude-result-prefixes="#all"
                version="2.0">

  <!--xsl:import href="../../ISM/IC-ISM-SecurityBanner.xsl"/>
  <xsl:import href="../../ISM/IC-ISM-PortionMark.xsl"/-->

  <!-- DAGON: test if saxon-he not supporting cast to list has to do with new ISM XSLs -->
  <xsl:import href="../../ISM-Old/IC-ISM-SecurityBanner.xsl"/>
  <xsl:import href="../../ISM-Old/IC-ISM-PortionMark.xsl"/>

  <xd:doc scope="stylesheet">
    <xd:desc>
      <xd:p>
        <xd:b>Created on:</xd:b> Sep 13, 2017</xd:p>
      <xd:p>
        <xd:b>Author:</xd:b> IC-CIO</xd:p>
      <xd:p>Convert a DED.xml instance document to Docbook for Rendering</xd:p>
    </xd:desc>
  </xd:doc>

  <xsl:output method="xml" indent="yes"/>

  <xd:doc>
    <xd:desc>Create the docbook book that is the data element dictionary</xd:desc>
  </xd:doc>
  <xsl:template match="ded:DataElementDictionary">
    <d:book version="5.0-ExtendForISM" xml:lang="en">

      <d:info>
<!-- <d:author>
  <d:personname>
    <d:surname>
      <xsl:value-of select="cve:IRM/cve:Source"/>
    </d:surname>
  </d:personname>
</d:author-->
      <d:releaseinfo>
        <xsl:apply-templates select="//arh:Security" mode="ism:banner"/>
      </d:releaseinfo>

      <!-- <d:mediaobject xml:id="image_odni">
      <d:imageobject>
```

```

        <d:imagedata fileref="../../../common/images/odni.jpg" width="1.59in" align="center"/>
    </d:imageobject>
</d:mediaobject>-->

<d:title>
    <xsl:apply-templates select="//ded:DedTitle" mode="ism:portionmark"/>Data Element Dictionary for <xsl:value-of select="//ded:DedTitle"/>
</d:title>
    <!--<d:edition>Version <xsl:value-of select="$spec.version"/></d:edition>-->
<d:author>
    <d:orgname>Prepared by <xsl:value-of select="//edh:ResponsibleEntity[@edh:role='Custodian']/edh:Organization"/>
    </d:orgname>
</d:author>

    <!--<d:pubdate>DraftDateReplaceMe</d:pubdate>
<d:date>ReleaseDateReplaceMe</d:date>
<d:address>Distribution Notice:
    This document has been approved for Public Release and is available for use without restriction.
</d:address>
    <d:releaseinfo>distFooterReplace</d:releaseinfo>-->
</d:info>
    <d:chapter>
        <d:title>Data Elements</d:title>
        <xsl:for-each select="./ded:DataElement">
            <xsl:sort select="ded:ElementName" order="ascending"/>
            <xsl:apply-templates select="current()"/>
        </xsl:for-each>
    </d:chapter>
    <xsl:if test="./ded:Relationship">
        <d:chapter>
            <d:title>Relationships</d:title>
            <xsl:for-each select="./ded:Relationship">
                <xsl:sort select="ded:RelationshipName" order="ascending"/>
                <xsl:apply-templates select="current()"/>
            </xsl:for-each>
        </d:chapter>
    </xsl:if>

    <d:index/>
</d:book>
</xsl:template>

<xd:doc>
    <xd:desc>Create the DataElement section</xd:desc>
</xd:doc>
<xsl:template match="ded:DataElement">
    <d:section>
        <d:title>
            <xsl:apply-templates select="ded:ElementName" mode="ism:portionmark"/>
            <xsl:value-of select="ded:ElementName"/>
        </d:title>
        <d:table frame="all" rowsep="1" colsep="1" xml:id="{ded:ElementName}">
            <d:title>
                <xsl:apply-templates select="ded:ElementName" mode="ism:portionmark"/>
            </d:title>
        </d:table>
    </d:section>
</xsl:template>

```

```

        <xsl:value-of select="ded:ElementName"/>
    </d:title>
    <d:indexterm>
        <d:primary>
            <xsl:value-of select="ded:ElementName"/>
        </d:primary>
    </d:indexterm>
    <d:tgroup cols="2" align="left">
        <d:colspec colname="c1" colnum="1" colwidth="1*" />
        <d:colspec colname="c2" colnum="2" colwidth="3*" />
        <d:thead>
            <d:row>
                <d:entry namest="c1" nameend="c2" align="center">
                    <d:para>
                        <xsl:apply-templates select="ded:ElementName" mode="ism:portionmark"/>The <xsl:value-of select="ded:ElementName"/> table is <xsl:apply-templates
select="." mode="ism:banner"/>
                    </d:para>
                </d:entry>
            </d:row>
            <d:row>
                <d:entry>
                    <d:para>DED Field Name</d:para>
                </d:entry>
                <d:entry>
                    <d:para>Value</d:para>
                </d:entry>
            </d:row>
        </d:thead>
        <d:tbody>
            <xsl:apply-templates select="./*">
                <xsl:sort select="local-name()" />
            </xsl:apply-templates>
        </d:tbody>
    </d:tgroup>
</d:table>
</d:section>

</xsl:template>

<xd:doc>
    <xd:desc>Create the Relationship section</xd:desc>
</xd:doc>
<xsl:template match="ded:Relationship" priority="5">
    <d:section>
        <d:title>
            <xsl:apply-templates select="ded:RelationshipName" mode="ism:portionmark"/>
            <xsl:value-of select="ded:RelationshipName"/>
        </d:title>
        <d:table frame="all"
            rowsep="1"
            colsep="1"
            xml:id="{ded:RelationshipName}">
            <d:title>
```

```

        <xsl:apply-templates select="ded:RelationshipName" mode="ism:portionmark"/>
        <xsl:value-of select="ded:RelationshipName"/>
    </d:title>
    <d:indexterm>
        <d:primary>
            <xsl:value-of select="ded:RelationshipName"/>
        </d:primary>
    </d:indexterm>
    <d:tgroup cols="2" align="left">
        <d:colspec colname="c1" colnum="1" colwidth="1*" />
        <d:colspec colname="c2" colnum="2" colwidth="3*" />
        <d:thead>
            <d:row>
                <d:entry namest="c1" nameend="c2" align="center">
                    <d:para>
                        <xsl:apply-templates select="ded:RelationshipName" mode="ism:portionmark"/>The <xsl:value-of select="ded:RelationshipName"/> table is <xsl:apply-
templates select="." mode="ism:banner"/>
                    </d:para>
                </d:entry>
            </d:row>
            <d:row>
                <d:entry>
                    <d:para>DED Field Name</d:para>
                </d:entry>
                <d:entry>
                    <d:para>Value</d:para>
                </d:entry>
            </d:row>
        </d:thead>
        <d:tbody>
            <xsl:apply-templates select="/*">
                <xsl:sort select="local-name()" />
            </xsl:apply-templates>
        </d:tbody>
    </d:tgroup>
</d:table>
</d:section>

</xsl:template>

<xd:doc>
    <xd:desc>Create a table row for each child of a DataElement or Relationship</xd:desc>
</xd:doc>
<xsl:template match="ded:*[parent::ded:DataElement or parent::ded:Relationship]">
    <d:row>
        <d:entry>
            <d:para>
                <xsl:value-of select="local-name()" />
            </d:para>
        </d:entry>
        <d:entry>
            <d:para>
                <xsl:choose>
                    <xsl:when test="@ism:*">
```

```

        <xsl:apply-templates select="." mode="ism:portionmark"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>(U) </xsl:text>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:value-of select="."/>
  </d:para>
</d:entry>
</d:row>
</xsl:template>

```

```

</xsl:stylesheet>

```

2.2 - Xsd-to-Ded.xsl

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xd="http://www.oxygenxml.com/ns/doc/xsl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:ded="urn:us:gov:ic:ded"
    xmlns:arh="urn:us:gov:ic:arh"
    xmlns:edh="urn:us:gov:ic:edh"
    xmlns:icid="urn:us:gov:ic:id"
    xmlns:usagency="urn:us:gov:ic:usagency"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xhtml="http://www.w3.org/1999/xhtml-StopBrowserRendering"
    exclude-result-prefixes="xs xd xhtml"
    version="2.0">
  <xd:doc scope="stylesheet">
    <xd:desc>
      <xd:p>
        <xd:b>Created on:</xd:b> Sep 12, 2017</xd:p>
      <xd:p>
        <xd:b>Author:</xd:b>IC-CIO</xd:p>
      <xd:p>Given an XSD with annotation documentation markup using the attribute data-ded to identify DED elements create a DED XML file from it. </xd:p>
    </xd:desc>
  </xd:doc>
  <xsl:output indent="yes"/>

  <xsl:param name="edhDESVersion" select="201903"/>
  <xsl:param name="usagencyCESVersion" select="202111"/>
  <xsl:param name="icidDESVersion" select="1"/>
  <xsl:param name="dedDESVersion" select="201903"/>

  <xd:doc>
    <xd:desc>Start at the beginning of the schema and run only on documentation with DedTitle</xd:desc>
  </xd:doc>
  <xsl:template match="/">
    <xsl:processing-instruction name="xml-model">href="../../Schematron/IC-EDH/IC-EDH_XML.sch" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"</
xsl:processing-instruction>
    <xsl:processing-instruction name="xml-model">href="../../Schematron/ISM/ISM_XML.sch" type="application/xml" schematypens=http://purl.oclc.org/dsdl/schematron"</
xsl:processing-instruction>
    <xsl:processing-instruction name="xml-model">href="../../Schematron/USAgency/USAgency_XML.sch" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"</
xsl:processing-instruction>
    <xsl:processing-instruction name="xml-model">href="../../Schematron/DED/DED_XML.sch" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"</
xsl:processing-instruction>
    <xsl:processing-instruction name="xml-model">href="../../Schematron/IC-ID/IC-ID_XML.sch" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"</
xsl:processing-instruction>

    <xsl:apply-templates select="//xsd:documentation[descendant::xhtml:p/@data-ded='ded:DedTitle']"/>
  </xsl:template>

  <xd:doc>
    <xd:desc>This is the "master" documentation it is the start of everything and the DED information on this entire DED.</xd:desc>
  </xd:doc>
  <xsl:template match="xsd:documentation[descendant::xhtml:div/@data-ded='ded:DataElementDictionary']">
    <ded:DataElementDictionary ded:DESVersion="{ $dedDESVersion }">
```

```

                icid:DESVersion="{ $icidDESVersion}"
                xsi:schemaLocation="urn:us:gov:ic:ded ../../Schema/DED/DED.xsd">
<ded:DedHeader>
  <edh:Edh edh:DESVersion="{ $edhDESVersion}"
    usagency:CESVersion="{ $usagencyCESVersion}">
    <icid:Identifier>
      <xsl:value-of select="normalize-space(substring-after(descendant::xhtml:p[@data-ded='ded:DedId'], ' '))"/>
    </icid:Identifier>
    <edh:DataItemCreateDateTime>
      <xsl:value-of select="normalize-space(substring-after(descendant::xhtml:p[@data-ded='ded:DedDate'], ' '))"/>
    </edh:DataItemCreateDateTime>
    <edh:ResponsibleEntity edh:role="Custodian">
      <edh:Country>USA</edh:Country>
      <edh:Organization>
        <xsl:value-of select="concat('USA.', normalize-space(substring-after(descendant::xhtml:p[@data-ded='ded:DedResponsibleAgency'], ' ')))"/>
      </edh:Organization>
    </edh:ResponsibleEntity>
    <arh:Security>
      <xsl:copy-of select="./descendant::xhtml:div[@data-ded='ded:DataElementDictionary']/@ism:*/>
      <xsl:if test="./descendant::xhtml:p[@data-ded='ded:DedDistroStatement']">
        <ism:NoticeList>
          <xsl:copy-of select="./descendant::xhtml:p[@data-ded='ded:DedDistroStatement']/@ism:*/>
          <ism:Notice ism:unregisteredNoticeType="distroStatement">
            <xsl:copy-of select="./descendant::xhtml:p[@data-ded='ded:DedDistroStatement']/@ism:*/>
            <ism:NoticeText>
              <xsl:copy-of select="./descendant::xhtml:p[@data-ded='ded:DedDistroStatement']/@ism:*/>
              <xsl:value-of select="normalize-space(substring-after(./descendant::xhtml:p[@data-ded='ded:DedDistroStatement'], ' '))"/>
            </ism:NoticeText>
          </ism:Notice>
        </ism:NoticeList>
      </xsl:if>
    </arh:Security>
  </edh:Edh>

  <xsl:apply-templates select="descendant::xhtml:p[@data-ded and not(@data-ded='ded:DedId' or @data-ded='ded:DedDate' or @data-ded='ded:DedDistroStatement' or @data-
ded='ded:DedResponsibleAgency')]" />
</ded:DedHeader>

  <xsl:apply-templates select="//xsd:documentation[descendant::xhtml:div/@data-ded='ded:DataElement']">
</xsl:apply-templates>
</ded:DataElementDictionary>
</xsl:template>

<xd:doc>
  <xd:desc>Create DataElements for each part of schema that is defined as such in documentation</xd:desc>
</xd:doc>
<xsl:template match="xsd:documentation[descendant::xhtml:div/@data-ded='ded:DataElement']">
  <ded:DataElement>
    <xsl:copy-of select="./descendant::xhtml:div[@data-ded='ded:DataElement']/@ism:*/>
    <xsl:apply-templates select="descendant::xhtml:p[@data-ded]" />
  </ded:DataElement>
</xsl:template>

```

```
<xd:doc>
  <xd:desc>Process schema documentation xhtml:p tags that have specific ded documentation components.</xd:desc>
</xd:doc>
<xsl:template match="xhtml:p[@data-ded]|xhtml:span[@data-ded]">
  <xsl:element name="{@data-ded}" namespace="urn:us:gov:ic:ded">
    <xsl:if test="not(@data-ded='ded:ElementId' or @data-ded='ded:DedVersion' or @data-ded='ded:DedShortName' or @data-ded='ded:DedSubjectType')">
      <xsl:copy-of select="@ism:*/">
    </xsl:if>
    <xsl:choose>
      <xsl:when test="descendant::xhtml:span">
        <xsl:apply-templates select="descendant::xhtml:span[1]">
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select="text()[normalize-space()]" mode="renderElementContent">
          <xsl:with-param name="elementName" select="@data-ded"/>
        </xsl:apply-templates>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:element>
</xsl:template>

<xd:doc>
  <xd:desc>Process text content of documentation to append element name if applicable</xd:desc>
  <xd:param name="elementName">Name of element to be prefixed on text content.</xd:param>
</xd:doc>
<xsl:template match="text()" mode="renderElementContent">
  <xsl:param name="elementName"/>
  <xsl:choose>
    <xsl:when test="contains(.,$elementName)">
      <xsl:value-of select="normalize-space(substring-after(., concat($elementName,':')))">
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="."/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xd:doc>
  <xd:desc>If something was not explicitly matched by another template, ignore it</xd:desc>
</xd:doc>
<xsl:template match="*" priority="0"/>

</xsl:stylesheet>
```