



Guide to XSLs for IC-TDF

IC-TDF XSL Guide

Version 2021-NOV

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 - GenerateXSDChangeXSL.xsl	2
2.2 - IC-TDF-Changes.xml	9
2.3 - IC-TDF-Changes.xsl	11
2.4 - Identity.xsl	15
2.5 - StripComments.xsl	16
2.6 - Tdf_to_TdfIc.xsl	17
2.7 - UpdateXSDDocumentation.xsl	18

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for IC-TDF. This document provides XSL files developed for IC-TDF.

Chapter 2 - XSL Files

2.1 - GenerateXSDChangeXSL.xsl

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
               xmlns:ch="urn:x-us:gov:ic:tdf:changes"
               xmlns:xslo="dummy"
               exclude-result-prefixes="ch"
               version="2.0">

  <!--
    A stylesheet to convert XPath expressions of changes into a stylesheet
    to actually perform the changes on an input file.

    $Id: changes2xsl.xsl,v 1.4 2006/04/05 19:31:17 holmank Exp $
  -->

  <xsl:namespace-alias stylesheet-prefix="xslo" result-prefix="xsl"/>

  <xsl:output indent="yes" encoding="US-ASCII"/>

  <!--=====-->
  <!--start the result stylesheet-->

  <xsl:template match="ch:changes">
    <!-- NOTE: add namespaces that we need to propagate to xslo:stylesheet tag -->
    <xslo:stylesheet xmlns:edh="urn:us:gov:ic:edh"
                   xmlns:arh="urn:us:gov:ic:arh"
                   xmlns:rr="urn:us:gov:ic:revrecall">

      <!--reduce namespace declarations by doing this up front-->
      <xsl:copy-of select="namespace::*"/>
      <xsl:attribute name="version">2.0</xsl:attribute>
      <xsl:attribute name="exclude-result-prefixes">
        <xsl:for-each select="namespace::*[.='urn:x-us:gov:ic:tdf:changes']">
          <xsl:value-of select="name(.)"/>
        </xsl:for-each>
      </xsl:attribute>
      <xslo:output encoding="US-ASCII"/>
      <xsl:text>

</xsl:text>

      <xsl:comment>present documentation before input comments</xsl:comment>
      <xslo:template match="/">
        <xsl:if test="ch:documentation">
          <xslo:text xml:space="preserve">

</xslo:text>

          <xsl:comment>
            <xsl:value-of select="ch:documentation"/>
          </xsl:comment>
          <xslo:text xml:space="preserve">

</xslo:text>

        </xsl:if>
        <xslo:apply-templates/>
      </xslo:template>

    </xslo:stylesheet>
  </xsl:template>

</xsl:stylesheet>
```

```
</xsl:text>
    <xsl:text>
</xsl:text>
    <xsl:comment>assume comments in prologue start on new lines</xsl:comment>
    <xsl:template match="/comment(">
        <xsl:text xml:space="preserve">
</xsl:text>
        <xsl:copy/>
    </xsl:template>
    <xsl:text>
</xsl:text>
    <xsl:comment>assume document element not being matched; copy namespaces to reduce the need for declarations</xsl:comment>
    <xsl:text>
</xsl:text>
    <xsl:template match="/*">
        <xsl:text xml:space="preserve">
</xsl:text>
        <xsl:element name="{name(/*)}" namespace="{namespace-uri(/*)}">
            <xsl:copy-of select="( / | document(''))/*" namespace::*[.!='http://www.w3.org/1999/XSL/Transform' and .!
='urn:x-us:gov:ic:tdf:changes']"/>
            <xsl:apply-templates select="@*|node()"/>
        </xsl:element>
    </xsl:template>
    <xsl:text>
</xsl:text>
    <xsl:comment>Identity transform for all unmached leaf nodes</xsl:comment>
    <xsl:text>
</xsl:text>
    <xsl:template match="@*|text()|comment()|processing-instruction(">
        <xsl:copy/>
    </xsl:template>
    <xsl:text>
</xsl:text>
    <xsl:comment>Identity transform for all unmached elements</xsl:comment>
    <xsl:text>
</xsl:text>
    <xsl:template match="*">
        <xsl:element name="{name(.)}" namespace="{namespace-uri(.)}">
            <xsl:apply-templates select="@*|node()"/>
        </xsl:element>
    </xsl:template>
    <xsl:text>
</xsl:text>
    <xsl:comment>Using a template allows xml:space to be used in param</xsl:comment>
    <xsl:text>
</xsl:text>
    <xsl:template name="copy-content">
        <xsl:param name="rtf"/>
        <xsl:copy-of select="$rtf"/>
    </xsl:template>
    <xsl:text>
```

```

</xsl:text>
    <xsl:comment>Expose current element's start tag</xsl:comment>
    <xsl:text>

</xsl:text>

    <xsl:template name="show-old-start-tag">
        <xsl:comment>
            <xsl:text xml:space="preserve">Replacing:

                <xsl:apply-templates mode="expose" select=".">
                    <xsl:with-param name="suppress" select="true()"/>
                </xsl:apply-templates>
                <xsl:text xml:space="preserve">

            </xsl:comment>
        </xsl:template>
        <xsl:template name="show-old-element">
            <xsl:comment>
                <xsl:text xml:space="preserve">Replacing:

                    <xsl:apply-templates mode="expose" select="."/>
                    <xsl:text xml:space="preserve">

                </xsl:comment>
            </xsl:template>
            <xsl:template mode="expose" match="*">
                <xsl:param name="suppress" select="false()"/>
                <xsl:value-of disable-output-escaping="yes" select="concat('&lt;','&lt;',name(.))"/>
                <xsl:for-each select="@*">
                    <xsl:value-of select="concat(' ','&lt;',name(.),'&lt;','&lt;',name(.),'&lt;','&lt;',name(.))"/>
                </xsl:for-each>
                <xsl:if test="not(node())"></xsl:if>
                <xsl:text>&lt;</xsl:text>
                <xsl:if test="not( $suppress ) and node()">
                    <xsl:apply-templates mode="expose"/>
                    <xsl:value-of disable-output-escaping="yes" select="concat('&lt;','&lt;',name(.),'&lt;','&lt;',name(.))"/>
                </xsl:if>
            </xsl:template>
            <xsl:template mode="expose" match="comment()">
                <xsl:value-of disable-output-escaping="yes" select="concat('&lt;!--','&lt;',name(.),'&lt;!--')"/>
            </xsl:template>
            <xsl:template mode="expose" match="processing-instruction()">
                <xsl:value-of disable-output-escaping="yes" select="concat('&lt;?','&lt;',name(.),'&lt;?','&lt;',name(.))"/>
            </xsl:template>

            <xsl:text>

        </xsl:text>

        <xsl:comment>Specific node changes follow</xsl:comment>
        <xsl:text>

    </xsl:text>

    <xsl:apply-templates/>
</xsl:stylesheet>
</xsl:template>

<!--=====

```

```
<!--copying the source tree content into the output stylesheet as stylesheet
  directives; requires preserving comments using a comment instruction-->

<xsl:template mode="copy-content" match="comment()" priority="2">
  <xsl:comment>
    <xsl:value-of select="."/>
  </xsl:comment>
</xsl:template>

  <xsl:template mode="copy-content" match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates mode="copy-content" select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

  <!--=====-->
<!--different directives-->

<!--documentation already handled-->
<xsl:template match="ch:documentation"/>

  <!--add content after the element being matched-->
<xsl:template match="ch:injectAfter">
  <xsl:call-template name="checkAttributes"/>
  <xsl:comment>injectAfter</xsl:comment>
  <xsl:template match="{@ch:match}">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
    <xsl:call-template name="copy-content">
      <xsl:with-param name="rtf">
        <xsl:attribute name="xml:space">preserve</xsl:attribute>
        <xsl:apply-templates mode="copy-content" select="node()"/>
      </xsl:with-param>
    </xsl:call-template>
  </xsl:template>
</xsl:template>

  <!--add content before the element being matched-->
<xsl:template match="ch:injectBefore">
  <xsl:call-template name="checkAttributes"/>
  <xsl:comment>injectBefore</xsl:comment>
  <xsl:template match="{@ch:match}">
    <xsl:call-template name="copy-content">
      <xsl:with-param name="rtf">
        <xsl:attribute name="xml:space">preserve</xsl:attribute>
        <xsl:apply-templates mode="copy-content" select="node()"/>
      </xsl:with-param>
    </xsl:call-template>
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>
</xsl:template>
```



```

        <!--add content at the beginning of the element being matched-->
<xsl:template match="ch:injectStart">
    <xsl:call-template name="checkAttributes"/>
    <xsl:comment>injectStart</xsl:comment>
    <xsl:template match="{@ch:match}">
        <xsl:copy>
            <xsl:apply-templates select="@*" />
            <xsl:call-template name="copy-content">
                <xsl:with-param name="rtf">
                    <xsl:attribute name="xml:space">preserve</xsl:attribute>
                    <xsl:apply-templates mode="copy-content" select="node()" />
                </xsl:with-param>
            </xsl:call-template>
            <xsl:apply-templates select="node()" />
        </xsl:copy>
    </xsl:template>
</xsl:template>

        <!--add content at the end of the element being matched-->
<xsl:template match="ch:injectEnd">
    <xsl:call-template name="checkAttributes"/>
    <xsl:comment>injectEnd</xsl:comment>
    <xsl:template match="{@ch:match}">
        <xsl:copy>
            <xsl:apply-templates select="@*|node()" />
            <xsl:call-template name="copy-content">
                <xsl:with-param name="rtf">
                    <xsl:attribute name="xml:space">preserve</xsl:attribute>
                    <xsl:apply-templates mode="copy-content" select="node()" />
                </xsl:with-param>
            </xsl:call-template>
        </xsl:copy>
    </xsl:template>
</xsl:template>

        <!--add attributes to the element being matched-->
<xsl:template match="ch:injectAttributes">
    <xsl:call-template name="checkAttributes"/>
    <xsl:comment>injectAttribute</xsl:comment>
    <xsl:template match="{@ch:match}">
        <xsl:call-template name="show-old-start-tag" />
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf">
                <xsl:attribute name="xml:space">preserve</xsl:attribute>
                <xsl:apply-templates mode="copy-content" select="text()|comment()" />
            </xsl:with-param>
        </xsl:call-template>
        <xsl:copy>
            <xsl:apply-templates select="@*" />
            <xsl:for-each select="@*[count(../@ch:* | .) != count(../@ch:*)]">
                <xsl:attribute name="{name(.)}" namespace="{namespace-uri(.)}">
                    <xsl:value-of select="."/>
                </xsl:attribute>
            </xsl:for-each>
        </xsl:copy>
    </xsl:template>
</xsl:template>

```

```

        </xsl:for-each>
        <xsl:apply-templates select="node()" />
    </xsl:copy>
</xsl:template>
</xsl:template>

    <!--add namespace to the element being matched-->
<!-- <xsl:template match="ch:injectNamespaces">
    <xsl:call-template name="checkAttributes" />
    <xsl:comment>injectNamespace</xsl:comment>
    <xsl:template match="{@ch:match}">
        <xsl:call-template name="show-old-start-tag" />
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf">
                <xsl:attribute name="xml:space">preserve</xsl:attribute>
                <xsl:apply-templates mode="copy-content" select="text()|comment()" />
            </xsl:with-param>
        </xsl:call-template>
    <xsl:copy>
        <xsl:apply-templates select="@*" />
        <xsl:for-each select="@*[count(../@ch:* | .) != count(../@ch:* )]">
            <xsl:namespace name="{name(.)}" select="'{.}'" />
        </xsl:for-each>
        <xsl:apply-templates select="node()" />
    </xsl:copy>
</xsl:template>
</xsl:template>-->

<!--replace the element being matched-->
<xsl:template match="ch:replaceElement">
    <xsl:call-template name="checkAttributes" />
    <xsl:comment>replaceElement</xsl:comment>
    <xsl:template match="{@ch:match}">
        <xsl:call-template name="show-old-element" />
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf">
                <xsl:attribute name="xml:space">preserve</xsl:attribute>
                <xsl:apply-templates mode="copy-content" select="node()" />
            </xsl:with-param>
        </xsl:call-template>
    </xsl:template>
</xsl:template>

    <!--report missing required attribute-->
<xsl:template name="checkAttributes">
    <xsl:if test="not(@ch:match)">
        <xsl:message terminate="yes">
            <xsl:text>Missing {urn:x-us:gov:ic:tdf:changes}match= for </xsl:text>
            <xsl:text/>element: <xsl:value-of select="name(.)" />
        </xsl:message>
    </xsl:if>
</xsl:template>

    <!--unexpected element-->
```

```
<xsl:template match="*">
  <xsl:message terminate="yes">
    <xsl:text/>Unexpected element: <xsl:value-of select="name(.)"/>
  </xsl:message>
</xsl:template>

</xsl:stylesheet>
```

2.2 - IC-TDF-Changes.xml

```
<changes xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xmlns="urn:x-us:gov:ic:tdf:changes"
          xmlns:xhtml="http://www.w3.org/1999/xhtml-StopBrowserRendering"
          xmlns:ch="urn:x-us:gov:ic:tdf:changes">

  <!-- The configuration file is manually edited. The documentation text below is used in the output which is a generated XSL
        for turning BASE-TDF into IC-TDF. -->
  <documentation>
    This file is auto-generated, do not edit this file directly.

    These are the changes needed to update BASE-TDF into IC-TDF.
    All changes are marked "Ic-Tdf:" with comments.

    IC-TDF-Changes.xml
  </documentation>

  <!-- INFO: new feature causes null pointer exception so any namespaces needed are directly added in GenerateXSDChangeXSL.xsl
        on the <xsl:stylesheet> -->
  <!-- <injectNamespaces
    ch:match="xs:schema[@targetNamespace='urn:us:gov:ic:tdf']"
    edh="urn:us:gov:ic:edh" arh="urn:us:gov:ic:arh" rr="urn:us:gov:ic:revrecall">
    <!--\- Ic-Tdf: Add the EDH, ARH, RevRecall namespaces -\->
  </injectNamespaces>
  -->

  <injectAttributes ch:match="xs:import[@namespace='urn:us:gov:ic:cenum:tdf:signaturealgorithm']"
                   schemaLocation="../../BASE-TDF/CVEGenerated/CVEnumTDFSignatureAlgorithm.xsd">
    <!-- Ic-Tdf: Update import path to BASE-TDF CVEGenerated for CVEnumTDFHashAlgorithm -->
  </injectAttributes>

  <injectAttributes ch:match="xs:import[@namespace='urn:us:gov:ic:cenum:tdf:hashalgorithm']"
                   schemaLocation="../../BASE-TDF/CVEGenerated/CVEnumTDFHashAlgorithm.xsd">
    <!-- Ic-Tdf: Update import path to BASE-TDF CVEGenerated for CVEnumTDFSignatureAlgorithm -->
  </injectAttributes>

  <injectAttributes ch:match="xs:import[@namespace='urn:us:gov:ic:cenum:tdf:state']"
                   schemaLocation="../../BASE-TDF/CVEGenerated/CVEnumTDFAppliesToState.xsd">
    <!-- Ic-Tdf: Update import path to BASE-TDF CVEGenerated for CVEnumTDFAppliesToState -->
  </injectAttributes>

  <injectAttributes ch:match="xs:attribute[@name='version']/xs:simpleType/xs:restriction/xs:pattern"
                   value="[0-9]{6}(\.[0-9]{6})?\-IC\-TDF\[0-9]{6}(\.[0-9]{6})?(\-\.{1,23})?">
    <!-- Ic-Tdf: Update tdf:version regex to be for IC-TDF customization -->
  </injectAttributes>

  <injectBefore ch:match="xs:element[@name='TrustedDataCollection']">
  <!-- Ic-Tdf: Add EDH, ARH, RevRecall schema imports -->
  <xs:import namespace="urn:us:gov:ic:edh" schemaLocation="../../IC-EDH/IC-EDH.xsd"/>
    <xs:import namespace="urn:us:gov:ic:arh" schemaLocation="../../ISM/IC-ARH.xsd"/>
    <xs:import namespace="urn:us:gov:ic:revrecall"
              schemaLocation="../../RevRecall/RevRecall_XML.xsd"/>
  </injectBefore>
```

```

        <replaceElement ch:match="xs:complexType[@name='StatementMetadataType']/xs:sequence/xs:any">
<!-- Ic-Tdf: Update xs:any to be xs:choice with EDH and ARH elements. -->
        <xs:choice>
            <xs:element ref="edh:Edh" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="edh:ExternalEdh" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="arh:Security" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="arh:ExternalSecurity" minOccurs="1" maxOccurs="1"/>
        </xs:choice>
    </replaceElement>

    <replaceElement ch:match="xs:complexType[@name='HandlingStatementType']/xs:sequence/xs:any">
<!-- Ic-Tdf: Update xs:any to be xs:choice with EDH and RevRecall elements. -->
    <xs:choice>
        <xs:element ref="edh:Edh" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="edh:ExternalEdh" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="rr:RevisionRecall" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
</replaceElement>

    <injectAttributes ch:match="xs:element[@name='HandlingAssertion']" minOccurs="1">
<!-- Ic-Tdf: Update minOccurs to 1 on HandlingAssertion. -->
</injectAttributes>

</changes>

```

2.3 - IC-TDF-Changes.xsl

```
<xsl:stylesheet xmlns:edh="urn:us:gov:ic:edh"
                xmlns:arh="urn:us:gov:ic:arh"
                xmlns:rr="urn:us:gov:ic:revrecall"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns="urn:x-us:gov:ic:tdf:changes"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns:xhtml="http://www.w3.org/1999/xhtml-StopBrowserRendering"
                xmlns:ch="urn:x-us:gov:ic:tdf:changes"
                version="2.0"
                exclude-result-prefixes="ch">
  <xsl:output encoding="US-ASCII"/>
  <!--present documentation before input comments--><xsl:template match="/">
    <xsl:text xml:space="preserve">

</xsl:text>
    <xsl:comment>
      This file is auto-generated, do not edit this file directly.

      These are the changes needed to update BASE-TDF into IC-TDF.
      All changes are marked "Ic-Tdf:" with comments.

      IC-TDF-Changes.xml
    </xsl:comment>
    <xsl:text xml:space="preserve">

</xsl:text>
    <xsl:apply-templates/>
  </xsl:template>
  <!--assume comments in prologue start on new lines--><xsl:template match="/comment(">
    <xsl:text xml:space="preserve">

</xsl:text>
    <xsl:copy/>
  </xsl:template>
  <!--assume document element not being matched; copy namespaces to reduce the need for declarations--><xsl:template match="/*">
    <xsl:text xml:space="preserve">

</xsl:text>
    <xsl:element name="{name(/*)}" namespace="{namespace-uri(/*)}">
      <xsl:copy-of select="( / | document(''))/*" namespace::*[.!='http://www.w3.org/1999/XSL/Transform' and .!='urn:x-us:gov:ic:tdf:changes']"/>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:element>
  </xsl:template>
  <!--Identity transform for all unmached leaf nodes--><xsl:template match="@*|text()|comment()|processing-instruction(">
    <xsl:copy/>
  </xsl:template>
  <!--Identity transform for all unmached elements--><xsl:template match="*">
    <xsl:element name="{name(.)}" namespace="{namespace-uri(.)}">
      <xsl:apply-templates select="@*|node()"/>
    </xsl:element>
  </xsl:template>
  <!--Using a template allows xml:space to be used in param--><xsl:template name="copy-content">
    <xsl:param name="rtf"/>
    <xsl:copy-of select="$rtf"/>
  </xsl:template>
```

```
<!--Expose current element's start tag--><xsl:template name="show-old-start-tag">
  <xsl:comment>
    <xsl:text xml:space="preserve">Replacing:
</xsl:text>

    <xsl:apply-templates mode="expose" select=".">
      <xsl:with-param name="suppress" select="true()"/>
    </xsl:apply-templates>
    <xsl:text xml:space="preserve">

  </xsl:comment>
</xsl:template>
<xsl:template name="show-old-element">
  <xsl:comment>
    <xsl:text xml:space="preserve">Replacing:
</xsl:text>

    <xsl:apply-templates mode="expose" select="."/>
    <xsl:text xml:space="preserve">

  </xsl:comment>
</xsl:template>
<xsl:template mode="expose" match="*">
  <xsl:param name="suppress" select="false()"/>
  <xsl:value-of disable-output-escaping="yes" select="concat('&lt;','&lt;',name())"/>
  <xsl:for-each select="@*">
    <xsl:value-of select="concat(' ',name().,'=&#34;','&#34;')"/>
  </xsl:for-each>
  <xsl:if test="not(node())"></xsl:if>
  <xsl:text>&gt;</xsl:text>
  <xsl:if test="not( $suppress ) and node()">
    <xsl:apply-templates mode="expose"/>
    <xsl:value-of disable-output-escaping="yes" select="concat('&lt;','&lt;',name().,'&gt;')"/>
  </xsl:if>
</xsl:template>
<xsl:template mode="expose" match="comment()">
  <xsl:value-of disable-output-escaping="yes" select="concat('&lt;!--','&lt;',name().,'--&gt;')"/>
</xsl:template>
<xsl:template mode="expose" match="processing-instruction()">
  <xsl:value-of disable-output-escaping="yes" select="concat('&lt;?','&lt;',name().,'?&gt;')"/>
</xsl:template>
<!--Specific node changes follow--><!--injectAttribute--><xsl:template match="xs:import[@namespace='urn:us:gov:ic:cvenum:tdf:signaturealgorithm']">
  <xsl:call-template name="show-old-start-tag"/>
  <xsl:call-template name="copy-content">
    <xsl:with-param name="rtf" xml:space="preserve">
    <xsl:comment> Ic-Tdf: Update import path to BASE-TDF CVEGenerated for CVEnumTDFHashAlgorithm </xsl:comment>
  </xsl:with-param>
  </xsl:call-template>
  <xsl:copy>
    <xsl:apply-templates select="@*">
      <xsl:attribute name="schemaLocation" namespace="">../BASE-TDF/CVEGenerated/CVEnumTDFSignatureAlgorithm.xsd</xsl:attribute>
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>
<!--injectAttribute--><xsl:template match="xs:import[@namespace='urn:us:gov:ic:cvenum:tdf:hashalgorithm']">
  <xsl:call-template name="show-old-start-tag"/>
```

```

        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf" xml:space="preserve">
                <xsl:comment> Ic-Tdf: Update import path to BASE-TDF CVEGenerated for CVEEnumTDFSignatureAlgorithm </xsl:comment>
            </xsl:with-param>
        </xsl:call-template>
        <xsl:copy>
            <xsl:apply-templates select="@*" />
            <xsl:attribute name="schemaLocation" namespace="">../BASE-TDF/CVEGenerated/CVEEnumTDFHashAlgorithm.xsd</xsl:attribute>
            <xsl:apply-templates select="node()" />
        </xsl:copy>
    </xsl:template>
    <!--injectAttribute--><xsl:template match="xs:import[@namespace='urn:us:gov:ic:cvenum:tdf:state']">
        <xsl:call-template name="show-old-start-tag" />
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf" xml:space="preserve">
                <xsl:comment> Ic-Tdf: Update import path to BASE-TDF CVEGenerated for CVEEnumTDFAppliesToState </xsl:comment>
            </xsl:with-param>
        </xsl:call-template>
        <xsl:copy>
            <xsl:apply-templates select="@*" />
            <xsl:attribute name="schemaLocation" namespace="">../BASE-TDF/CVEGenerated/CVEEnumTDFAppliesToState.xsd</xsl:attribute>
            <xsl:apply-templates select="node()" />
        </xsl:copy>
    </xsl:template>
    <!--injectAttribute--><xsl:template match="xs:attribute[@name='version']/xs:simpleType/xs:restriction/xs:pattern">
        <xsl:call-template name="show-old-start-tag" />
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf" xml:space="preserve">
                <xsl:comment> Ic-Tdf: Update tdf:version regex to be for IC-TDF customization </xsl:comment>
            </xsl:with-param>
        </xsl:call-template>
        <xsl:copy>
            <xsl:apply-templates select="@*" />
            <xsl:attribute name="value" namespace="">[0-9]{6}(\.[0-9]{6})?\-IC\-TDF\[0-9]{6}(\.[0-9]{6})?(\-.{1,23})?</xsl:attribute>
            <xsl:apply-templates select="node()" />
        </xsl:copy>
    </xsl:template>
    <!--injectBefore--><xsl:template match="xs:element[@name='TrustedDataCollection']">
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf" xml:space="preserve">
                <xsl:comment> Ic-Tdf: Add EDH, ARH, RevRecall schema imports </xsl:comment>
                <xs:import namespace="urn:us:gov:ic:edh" schemaLocation="../IC-EDH/IC-EDH.xsd"/>
                <xs:import namespace="urn:us:gov:ic:arh" schemaLocation="../ISM/IC-ARH.xsd"/>
                <xs:import namespace="urn:us:gov:ic:revrecall" schemaLocation="../RevRecall/RevRecall_XML.xsd"/>
            </xsl:with-param>
        </xsl:call-template>
        <xsl:copy>
            <xsl:apply-templates select="@*|node()" />
        </xsl:copy>
    </xsl:template>
    <!--replaceElement--><xsl:template match="xs:complexType[@name='StatementMetadataType']/xs:sequence/xs:any">
        <xsl:call-template name="show-old-element" />
        <xsl:call-template name="copy-content">
            <xsl:with-param name="rtf" xml:space="preserve">

```



```

<xsl:comment> Ic-Tdf: Update xs:any to be xs:choice with EDH and ARH elements. </xsl:comment>
  <xs:choice>
    <xs:element ref="edh:Edh" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="edh:ExternalEdh" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="arh:Security" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="arh:ExternalSecurity" minOccurs="1" maxOccurs="1"/>
  </xs:choice>
</xsl:with-param>
  </xsl:call-template>
</xsl:template>
<!--replaceElement--><xsl:template match="xs:complexType[@name='HandlingStatementType']/xs:sequence/xs:any">
  <xsl:call-template name="show-old-element"/>
  <xsl:call-template name="copy-content">
    <xsl:with-param name="rtf" xml:space="preserve">
<xsl:comment> Ic-Tdf: Update xs:any to be xs:choice with EDH and RevRecall elements. </xsl:comment>
<xs:choice>
  <xs:element ref="edh:Edh" minOccurs="1" maxOccurs="1"/>
  <xs:element ref="edh:ExternalEdh" minOccurs="1" maxOccurs="1"/>
  <xs:element ref="rr:RevisionRecall" minOccurs="0" maxOccurs="1"/>
</xs:choice>
</xsl:with-param>
  </xsl:call-template>
</xsl:template>
<!--injectAttribute--><xsl:template match="xs:element[@name='HandlingAssertion']">
  <xsl:call-template name="show-old-start-tag"/>
  <xsl:call-template name="copy-content">
    <xsl:with-param name="rtf" xml:space="preserve">
<xsl:comment> Ic-Tdf: Update minOccurs to 1 on HandlingAssertion. </xsl:comment>
</xsl:with-param>
  </xsl:call-template>
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:attribute name="minOccurs" namespace="">1</xsl:attribute>
    <xsl:apply-templates select="node()"/>
  </xsl:copy>
</xsl:template>
</xsl:stylesheet>

```

2.4 - Identity.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"
                exclude-result-prefixes="#all"
                version="2.0">
  <xd:doc scope="stylesheet">
    <xd:desc>
      <xd:p>
        <xd:b>Created on:</xd:b> Nov 27, 2019</xd:p>
      <xd:p>
        <xd:b>Author:</xd:b> dagon</xd:p>
      <xd:p>Identity XSL needed to fix xmlns=""</xd:p>
    </xd:desc>
  </xd:doc>

  <xsl:output method="xml" indent="yes" encoding="utf-8"/>
  <xsl:strip-space elements="*" />

  <xsl:template match="@*|node()">
    <xsl:copy copy-namespaces="yes">
      <xsl:apply-templates select="@*|node()" />
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

2.5 - StripComments.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"
                exclude-result-prefixes="xs xd"
                version="2.0">
  <xd:doc scope="stylesheet">
    <xd:desc>
      <xd:p>
        <xd:b>Created on:</xd:b> Mar 13, 2019</xd:p>
      <xd:p>
        <xd:b>Author:</xd:b> bob</xd:p>
      <xd:p>Strip comments and annotations out of xsd file to make the smallest simple file for a guard.</xd:p>
    </xd:desc>
  </xd:doc>

  <xsl:output method="xml" indent="yes" encoding="utf-8"/>
  <xsl:strip-space elements="*" />

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()" />
    </xsl:copy>
  </xsl:template>

  <xd:doc>
    <xd:desc>
      <xd:p>Strip out all comments, PIs and xs:annotations.</xd:p>
    </xd:desc>
  </xd:doc>
  <xsl:template match="comment()|processing-instruction()|xs:annotation" priority="5"/>
</xsl:stylesheet>
```

2.6 - Tdf_to_Tdflc.xsl

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns:xhtml="http://www.w3.org/1999/xhtml"
                xmlns:ntk="urn:us:gov:ic:ntk"
                xmlns:edh="urn:us:gov:ic:edh"
                xmlns:arh="urn:us:gov:ic:arh"
                exclude-result-prefixes="xs"
                version="2.0">

  <!-- ***** -->
  <!-- Identity template -->
  <!-- ***** -->
  <xsl:template match="@*|node()" name="identity">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()" />
    </xsl:copy>
  </xsl:template>

  <xsl:template match="xs:import">
    <xs:import namespace="urn:us:gov:ic:edh" schemaLocation="../IC-EDH/IC-EDH.xsd"/>
    <xs:import namespace="urn:us:gov:ic:arh" schemaLocation="../ISM/IC-ARH.xsd"/>
  </xsl:template>

  <xsl:template match="//xs:complexType[@name='KeyAccessType']/xs:any"/>

  <xsl:template match="//xs:complexType[@name='BindingInformationType']/xs:element[@ref='ds:KeyInfo']"/>

  <xsl:template match="//xs:element[@name='HandlingStatement']/xs:any">
    <xs:choice>
      <xs:element ref="edh:Edh" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="edh:ExternalEdh" minOccurs="1" maxOccurs="1"/>
    </xs:choice>
  </xsl:template>

  <xsl:template match="//xs:element[@name='StatementMetadata']/xs:any">
    <xs:choice>
      <xs:element ref="edh:Edh" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="edh:ExternalEdh" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="arh:Security" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="arh:ExternalSecurity" minOccurs="1" maxOccurs="1"/>
    </xs:choice>
  </xsl:template>

  <xsl:template match="//xs:element[@name='EncryptionInformation']/xs:any"/>

</xsl:stylesheet>
```

2.7 - UpdateXSDDocumentation.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:xhtml="http://www.w3.org/1999/xhtml-StopBrowserRendering"
                exclude-result-prefixes="xs xd"
                version="2.0">
  <xd:doc scope="stylesheet">
    <xd:desc>
      <xd:p>
        <xd:b>Created on:</xd:b>Nov 19, 2019</xd:p>
      <xd:p>
        <xd:b>Author:</xd:b> dagon</xd:p>
      <xd:p>Update the header and footer annotation documentation in the generated IC-TDF XSD.</xd:p>
    </xd:desc>
  </xd:doc>

  <xsl:output method="xml" indent="yes" encoding="utf-8"/>
  <xsl:strip-space elements="*" />

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()" />
    </xsl:copy>
  </xsl:template>

  <xsl:template match="/">
    <xsl:choose>
      <xsl:when test="not(../xs:annotation[../xhtml:a/@href = '../..Documents/BASE-TDF/DesBaseTDFXml.pdf'])">
        <xsl:call-template name="AbortDueToMissingHeaderAnnotations" />
      </xsl:when>
      <xsl:when test="not(../xs:annotation[../xhtml:h2/text() = 'Formal Change List'])">
        <xsl:call-template name="AbortDueToMissingFooterAnnotations" />
      </xsl:when>
      <xsl:otherwise/>
    </xsl:choose>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template name="AbortDueToMissingHeaderAnnotations">
    <xsl:message terminate="yes">
      <xsl:value-of select="'Missing Expected Header Annotations.'"/>
    </xsl:message>
  </xsl:template>

  <xsl:template name="AbortDueToMissingFooterAnnotations">
    <xsl:message terminate="yes">
      <xsl:value-of select="'Missing Expected Footer Annotations.'"/>
    </xsl:message>
  </xsl:template>

  <!-- Update annotation documentation header (description, introduction, implementation notes section)-->
  <xsl:template match="xs:annotation[../xhtml:a/@href = '../..Documents/BASE-TDF/DesBaseTDFXml.pdf']">
```

```
<xs:annotation>
  <xs:documentation>
    <xhtml:h1 ism:ownerProducer="USA" ism:classification="U">Intelligence Community
Technical Specification XML Data Encoding Specification for Trusted Data Format
(IC-TDF.XML)</xhtml:h1>
  </xs:documentation>
  <xs:documentation>
    <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Notices</xhtml:h2>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">Distribution Notice:
This document has been approved for Public Release and is available for use without restriction.
</xhtml:p>
  </xs:documentation>
  <xs:documentation>
    <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Description</xhtml:h2>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">W3C XML Schema for the XML Data
Encoding Specification for the intelligence version of Trusted Data Format
(IC-TDF.XML).</xhtml:p>
  </xs:documentation>
  <xs:documentation>
    <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Introduction</xhtml:h2>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">This XML Schema file is one
component of the XML Data Encoding Specification (DES). Please see the document titled<xhtml:i>
    <xhtml:a href="../../Documents/IC-TDF/DesTdfIcXml.pdf">XML Data Encoding
Specification for Intelligence Trusted Data Format</xhtml:a>
    </xhtml:i>for a complete description of the encoding as well as list of all
components.</xhtml:p>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">It is envisioned that this
schema or its components, as well as other parts of the DES may be overridden for
localized implementations. Therefore, permission to use, copy, modify and distribute
this XML Schema and the other parts of the DES for any purpose is hereby granted in
perpetuity.</xhtml:p>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">Please reference the preceding
two paragraphs in all copies or variations. The developers make no representation
about the suitability of the schema or DES for any purpose. It is provided "as is"
without expressed or implied warranty.</xhtml:p>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">If you modify this XML Schema in
any way label your schema as a variant of IC-TDF.XML.</xhtml:p>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">Please direct all questions, bug
reports,or suggestions for changes to the points of contact identified in the
document referenced above.</xhtml:p>
  </xs:documentation>
  <xs:documentation>
    <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Implementation Notes</xhtml:h2>
    <xhtml:p ism:ownerProducer="USA" ism:classification="U">The root elements for a TDF
instance are either: <xhtml:ul>
      <xhtml:li ism:ownerProducer="USA" ism:classification="U">
        <xhtml:a href="IC-TDF_xsd_Element_TrustedDataCollection.html#TrustedDataCollection">tdf:TrustedDataCollection</xhtml:a>
      </xhtml:li>
      <xhtml:li ism:ownerProducer="USA" ism:classification="U">
        <xhtml:a href="IC-TDF_xsd_Element_TrustedDataObject.html#TrustedDataObject">tdf:TrustedDataObject</xhtml:a>
      </xhtml:li>
    </xhtml:ul>
  </xhtml:p>
</xs:documentation>
```

```
<xs:documentation>
  <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Creators</xhtml:h2>
  <xhtml:p ism:ownerProducer="USA" ism:classification="U">Office of the Director of
National Intelligence Intelligence Community Chief Information Officer</xhtml:p>
</xs:documentation>
</xs:annotation>

</xsl:template>

<!-- Update annotation documentation footer (change history section) -->
<xsl:template match="xs:annotation[./xhtml:h2/text() = 'Formal Change List']">

  <xs:annotation>
    <xs:documentation>
      <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Formal Change List</xhtml:h2>
      <xhtml:table ism:ownerProducer="USA" ism:classification="U" id="ChangeHistory">
        <xhtml:caption>Change History</xhtml:caption>
        <xhtml:thead>
          <xhtml:tr>
            <xhtml:th>Version</xhtml:th>
            <xhtml:th>Date</xhtml:th>
            <xhtml:th>By</xhtml:th>
            <xhtml:th>Description</xhtml:th>
          </xhtml:tr>
        </xhtml:thead>
        <xhtml:tbody>
          <xhtml:tr>
            <xhtml:td>2017-JUL</xhtml:td>
            <xhtml:td>2017-04-27</xhtml:td>
            <xhtml:td>ODNI/OCIO/ICEA</xhtml:td>
            <xhtml:td>
              <xhtml:ul>
                <xhtml:li ism:ownerProducer="USA" ism:classification="U">
                  The schema change logs will no longer be maintained as of the 2017-JUL release.
                  The existing change logs will only serve as legacy information.
                  For changes to schema as of and after 2017-JUL, reference the change history in the DES.</xhtml:li>
                </xhtml:ul>
              </xhtml:td>
            </xhtml:tr>
            <xhtml:tr>
              <xhtml:td>2014-DEC</xhtml:td>
              <xhtml:td>2014-10-28</xhtml:td>
              <xhtml:td>Hodges, ODNI/CIO/ME/EI&amp;A</xhtml:td>
              <xhtml:td>Changed assertion group to be 1-to-1 instead of 1-to-Many for TDO which was missed previously.</xhtml:td>
            </xhtml:tr>
            <xhtml:tr>
              <xhtml:td>2014-DEC</xhtml:td>
              <xhtml:td>2014-12-04</xhtml:td>
              <xhtml:td>ODNI/OCIO/ICEA</xhtml:td>
              <xhtml:td>Enforced DESVersion with a pattern instead of a fixed value to allow extension versions</xhtml:td>
            </xhtml:tr>
            <xhtml:tr>
              <xhtml:td>2014-DEC</xhtml:td>
              <xhtml:td>2014-08-28</xhtml:td>
```

```
<xhtml:td>Lucas, ODNI/CIO/ME/EI&A</xhtml:td>
<xhtml:td>Updated to support Revision Recall Handling Assertion.</xhtml:td>
</xhtml:tr>
<xhtml:tr>
  <xhtml:td>2014-DEC</xhtml:td>
  <xhtml:td>2014-08-28</xhtml:td>
  <xhtml:td>Hodges, ODNI/CIO/ME/EI&A</xhtml:td>
  <xhtml:td>Added additional encryption fields for:
<xhtml:ul>
  <xhtml:li>Initialization Vector</xhtml:li>
  <xhtml:li>tweak</xhtml:li>
  <xhtml:li>nonce</xhtml:li>
  <xhtml:li>hash algorithm</xhtml:li>
  <xhtml:li>Mask Generation Function</xhtml:li>
  <xhtml:li>Addition Authentication Data</xhtml:li>
  <xhtml:li>authentication tag</xhtml:li>
  <xhtml:li>key encoding format</xhtml:li>
  <xhtml:li>Provable Data Possession wrapped keys</xhtml:li>
</xhtml:ul>
</xhtml:td>
</xhtml:tr>
<xhtml:tr>
  <xhtml:td>2014-DEC</xhtml:td>
  <xhtml:td>2014-08-28</xhtml:td>
  <xhtml:td>Hodges, ODNI/CIO/ME/EI&A</xhtml:td>
  <xhtml:td>Enforcing ordiner so Handling assertions comes first:
<xhtml:ul>
  <xhtml:li>Changed cardinality of assertion group to be 1-to-1 instead of 1-to-Many for TDC</xhtml:li>
  <xhtml:li>Changed cardinality of HandlingAssertions in assertionGroup to be 1-to-Many from 1-to-1</xhtml:li>
  <xhtml:li>Changed cardinality of HandlingAssertions in assertionGroup to be 0-to-Many from 1-to-1</xhtml:li>
</xhtml:ul>
</xhtml:td>
</xhtml:tr>
<xhtml:tr>
  <xhtml:td>3</xhtml:td>
  <xhtml:td>2013-05-27</xhtml:td>
  <xhtml:td>Clay, ODNI/CIO/ME/DIA</xhtml:td>
  <xhtml:td>Removed scope [TDO PAYL]</xhtml:td>
</xhtml:tr>
<xhtml:tr>
  <xhtml:td>2</xhtml:td>
  <xhtml:td>2012-11-12</xhtml:td>
  <xhtml:td>D'Ambra, ODNI/CIO/ME/DIA</xhtml:td>
  <xhtml:td>Added scopes [DESC_TDO], [DESC_PAYL] and [TDC_MEMBER] for use
within TDC Assertions</xhtml:td>
</xhtml:tr>
<xhtml:tr>
  <xhtml:td>2</xhtml:td>
  <xhtml:td>2012-11-12</xhtml:td>
  <xhtml:td>Gilsenan, ODNI/CIO/ME/DIA</xhtml:td>
  <xhtml:td>Added support for multiple binding by changing @max to "unbounded"
on Binding element in the BindingGroup</xhtml:td>
</xhtml:tr>
<xhtml:tr>
```



```

        <xhtml:td>2</xhtml:td>
        <xhtml:td>2012-11-11</xhtml:td>
        <xhtml:td>D'Ambra, ODNI/CIO/ME/DIA</xhtml:td>
        <xhtml:td>Changed @version to be fixed to 2</xhtml:td>
    </xhtml:tr>
</xhtml:tbody>
</xhtml:table>
</xs:documentation>
</xs:annotation>
</xsl:template>

</xsl:stylesheet>
```