



Guide to XSLs for BOE

BOE XSL Guide

Version 2018-APR

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 - Boe-to-Docbook.xsl	2

Chapter 1 - Introduction

1.1 - Purpose

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

Chapter 2 - XSL Files

2.1 - Boe-to-Docbook.xsl

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

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

    <xsl:output method="xml" encoding="UTF-8" indent="yes"/>

    <xsl:template match="/">
        <d:book version="5.0-variant urn:us:gov:ic:docbook-201804" 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 role="usa.nsi.banner">
                <xsl:apply-templates select="boe:BOE/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>Body of Evidence for <xsl:value-of select="//boe:SystemInformation/boe:SystemName"/>
            </d:title>
            <!--<d:edition>Version <xsl:value-of select="$spec.version"/></d:edition-->
<!--<d:author>
    <d:orgname>Prepared by the </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>

        <xsl:apply-templates select="* except arh:Security"/>
    </d:book>
```

```
</xsl:template>

<xsl:template match="boe:SystemInformation">
  <d:chapter>
    <d:title>(U) System Information</d:title>
    <xsl:apply-templates select=".*"/>
  </d:chapter>
</xsl:template>

<xsl:template match="boe:SSP">
  <d:chapter>
    <d:title>(U) System Security Plan</d:title>
    <xsl:apply-templates select=".*"/>
  </d:chapter>
</xsl:template>

<xsl:template match="boe:SAR">
  <d:chapter>
    <d:title>(U) Security Assessment Report</d:title>
    <xsl:apply-templates select=".*"/>
  </d:chapter>
</xsl:template>

<xsl:template match="boe:RAR">
  <d:chapter>
    <d:title>(U) Risk Assessment Report</d:title>
    <xsl:apply-templates select=".*"/>
  </d:chapter>
</xsl:template>

<xsl:template match="boe:POAM">
  <d:chapter>
    <d:title>(U) Plan of Action and Milestones</d:title>
    <xsl:apply-templates select=".*"/>
  </d:chapter>
</xsl:template>

<xsl:template match="boe:ADD">
  <d:chapter>
    <d:title>(U) Authorization Decision Document</d:title>
    <xsl:apply-templates select=".*"/>
  </d:chapter>
</xsl:template>

<!-- InformationTechnologyType requires special handling -->
<xsl:template match="boe:SystemInformation/boe:InformationTechnologyType"
  priority="5">
  <d:para>
    <d:emphasis role="bold">
      <xsl:apply-templates select="." mode="ism:portionmark"/>
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()"/>
      </xsl:call-template>
    </d:emphasis>
    <xsl:text>: </xsl:text>
  </d:para>
</xsl:template>
```

```

        </d:emphasis>
        <d:itemizedlist>
            <d:listitem>
                <d:para>Type of Service: <xsl:value-of select="@boe:typeOfService"/>
            </d:para>
            </d:listitem>
            <d:listitem>
                <d:para>Is a Standalone System:
                <xsl:call-template name="capitalizeFirstLetter">
                    <xsl:with-param name="string" select="@boe:isStandalone"/>
                </xsl:call-template>
            </d:para>
            </d:listitem>
        </d:itemizedlist>
    </d:para>
</xsl:template>

<!-- SecurityCategorization requires special handling -->
<xsl:template match="boe:SystemInformation/boe:SecurityCategorization" priority="5">
    <d:para>
        <d:emphasis role="bold">
            <xsl:apply-templates select="." mode="ism:portionmark"/>
            <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()"/>
            </xsl:call-template>
            <xsl:text>: </xsl:text>
        </d:emphasis>
        <d:itemizedlist>
            <xsl:for-each select="*">
                <d:listitem>
                    <xsl:call-template name="responseWithExplanation"/>
                </d:listitem>
            </xsl:for-each>
        </d:itemizedlist>
    </d:para>
</xsl:template>

<xsl:template match="boe:SystemInformation/boe:DataAuthorizationLevel" priority="5">
    <d:para>
        <d:emphasis role="bold">
            <xsl:apply-templates select="." mode="ism:portionmark"/>
            <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()"/>
            </xsl:call-template>
            <xsl:text>: </xsl:text>
        </d:emphasis>
        <d:itemizedlist>
            <xsl:for-each select="*">
                <d:listitem>
                    <xsl:apply-templates select="."/>
                </d:listitem>
            </xsl:for-each>
        </d:itemizedlist>
    </d:para>

```

```
</xsl:template>

<xsl:template match="boe:DataAuthorizationLevel/uias:Clearance" priority="5">
  <xsl:variable name="tempISM">
    <xsl:element name="temp">
      <xsl:attribute name="classification" namespace="urn:us:gov:ic:ism" select="."/>
      <xsl:attribute name="ownerProducer" namespace="urn:us:gov:ic:ism" select="'USA'"/>
    </xsl:element>
  </xsl:variable>
  <d:para>
    <d:emphasis role="bold">Highest Classification Level: </d:emphasis>
    <xsl:apply-templates select="$tempISM" mode="ism:banner"/>
  </d:para>
</xsl:template>

<xsl:template match="boe:DataAuthorizationLevel/uias:FineAccessControls | boe:DataAuthorizationLevel/uias:HandlingControls"
  priority="5">
  <d:para>
    <d:emphasis role="bold">
      <xsl:text>Authorized </xsl:text>
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()"/>
      </xsl:call-template>
      <xsl:text>: </xsl:text>
    </d:emphasis>
    <xsl:value-of select="text()"/>
  </d:para>
</xsl:template>

<!-- CryptographicKey requires special handling -->
<xsl:template match="boe:SSP/boe:CryptographicKey" priority="5">
  <d:para>
    <d:emphasis role="bold">
      <!--<xsl:apply-templates select="self::node()" mode="ism:portionmark"/>-->
      <xsl:text>(U) </xsl:text>
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()"/>
      </xsl:call-template>
      <xsl:text>: </xsl:text>
    </d:emphasis>
    <d:itemizedlist>
      <xsl:for-each select="/*">
        <d:listitem>
          <xsl:call-template name="responseWithExplanation">
            <xsl:with-param name="title">
              <xsl:choose>
                <xsl:when test="local-name()='FIPSVvalidatedUnclass'">
                  <xsl:text>FIPS Validated for Unclassified Information</xsl:text>
                </xsl:when>
                <xsl:when test="local-name()='FIPSVvalidatedCompartment'">
                  <xsl:text>FIPS Validated for Compartmented Information</xsl:text>
                </xsl:when>
                <xsl:when test="local-name()='NSAApprovedCrypto'">
                  <xsl:text>Uses NSA Approved Cryptography</xsl:text>
                </xsl:when>
              </xsl:choose>
            </xsl:with-param>
          </xsl:call-template>
        </d:listitem>
      </xsl:for-each>
    </d:itemizedlist>
  </d:para>
</xsl:template>
```



```

        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()"/>
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>
</xsl:with-param>
</xsl:call-template>
</d:listitem>
</xsl:for-each>
</d:itemizedlist>
</d:para>
</xsl:template>

<xsl:template match="boe:SSP/boe:ExternalSecurityServices" priority="5">
    <xsl:variable name="title">
        <xsl:call-template name="breakIntoWords">
            <xsl:with-param name="string" select="local-name()"/>
        </xsl:call-template>
    </xsl:variable>
    <d:para>
        <d:emphasis role="bold">
            <xsl:text>(U) </xsl:text>
            <xsl:value-of select="$title"/>
            <xsl:text>: </xsl:text>
        </d:emphasis>The following table is <xsl:apply-templates select="." mode="ism:banner"/>.</d:para>

        <d:table frame="all">
            <d:title>(U) <xsl:value-of select="$title"/>
            </d:title>
            <d:tgroup cols="4">
                <d:colspec colname="c1" colnum="1" colwidth="1.0*" />
                <d:colspec colname="c2" colnum="2" colwidth="1.0*" />
                <d:colspec colname="c3" colnum="3" colwidth="1.0*" />
                <d:colspec colname="c4" colnum="4" colwidth="3.0*" />
                <d:thead>
                    <d:row>
                        <d:entry>Service Name</d:entry>
                        <d:entry>Service Provider</d:entry>
                        <d:entry>Meets Organization Security Requirements?</d:entry>
                        <d:entry>Description</d:entry>
                    </d:row>
                </d:thead>
                <d:tbody>
                    <xsl:apply-templates select=".*" />
                </d:tbody>
            </d:tgroup>
        </d:table>
    </xsl:template>

    <xsl:template match="boe:SecurityService">
        <d:row>
```

```

    <d:entry>
      <xsl:value-of select="@boe:name" />
    </d:entry>
    <d:entry>
      <xsl:value-of select="@boe:provider" />
    </d:entry>
    <d:entry>
      <xsl:call-template name="capitalizeFirstLetter">
        <xsl:with-param name="string" select="@boe:meetsOrgSecurityRequirements" />
      </xsl:call-template>
    </d:entry>
    <d:entry>
      <xsl:apply-templates select="boe:Description" mode="ism:portionmark" />
      <xsl:apply-templates select="boe:Description" />
    </d:entry>
  </d:row>
</xsl:template>

<xsl:template match="boe:SSP/boe:KeyRoles" priority="5">
  <d:para>
    <d:emphasis role="bold">
      <xsl:apply-templates select="." mode="ism:portionmark" />
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()" />
      </xsl:call-template>
      <xsl:text>: </xsl:text>
    </d:emphasis>
    <d:itemizedlist>
      <xsl:for-each select="."/ *>
        <xsl:variable name="title">
          <xsl:choose>
            <xsl:when test="local-name()='Other'">
              <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string">
                  <xsl:call-template name="capitalizeFirstLetter">
                    <xsl:with-param name="string" select="boe:Role" />
                  </xsl:call-template>
                </xsl:with-param>
              </xsl:call-template>
            <xsl:otherwise>
              <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()" />
              </xsl:call-template>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:variable>
        <d:listitem>
          <d:para>
            <d:emphasis role="bold">
              <xsl:apply-templates select="." mode="ism:portionmark" />
              <xsl:value-of select="$title" />
            </d:emphasis>
          </d:para>
        </d:listitem>
      </xsl:for-each>
    </d:itemizedlist>
  </d:para>

```

```

        <xsl:for-each select="/* except boe:Role">
            <d:listitem>
                <xsl:call-template name="responseWithExplanation">
                    <xsl:with-param name="response" select="text()"/>
                    <xsl:with-param name="hasExplanation" select="false()"/>
                </xsl:call-template>
            </d:listitem>
        </xsl:for-each>
    </d:itemizedlist>
</d:para>
</d:listitem>
</xsl:for-each>
</d:itemizedlist>
</d:para>
</xsl:template>

<xsl:template match="boe:SSP/boe:SystemUserCategories" priority="5">
    <d:para>
        <d:emphasis role="bold">
            <xsl:apply-templates select="." mode="ism:portionmark"/>
            <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()"/>
            </xsl:call-template>
            <xsl:text>: </xsl:text>
        </d:emphasis>
        <d:itemizedlist>
            <xsl:apply-templates select="/*"/>
        </d:itemizedlist>
    </d:para>
</xsl:template>

<xsl:template match="boe:SystemUserCategory">
    <d:listitem>
        <d:para>
            <d:emphasis role="bold">
                <xsl:apply-templates select="." mode="ism:portionmark"/>
                <xsl:value-of select="@boe:userCategory"/>
                <xsl:text>: </xsl:text>
            </d:emphasis>
            <d:itemizedlist>
                <xsl:apply-templates select="/*"/>
            </d:itemizedlist>
        </d:para>
    </d:listitem>
</xsl:template>

<xsl:template match="boe:AccessDescription">
    <d:listitem>
        <xsl:call-template name="responseWithExplanation">
            <xsl:with-param name="response" select="text()"/>
            <xsl:with-param name="hasExplanation" select="false()"/>
        </xsl:call-template>
    </d:listitem>
</xsl:template>

```

```

        <!--<xsl:template match="boe:AssessmentDate">
<d:listitem>
  <xsl:call-template name="responseWithExplanation">
    <xsl:with-param name="response" select="text()"/>
    <xsl:with-param name="hasExplanation" select="false()"/>
  </xsl:call-template>
</d:listitem>
</xsl:template>-->

<xsl:template match="boe:UserConstraints">
  <d:listitem>
    <d:para>
      <d:emphasis role="bold">
        <xsl:call-template name="breakIntoWords">
          <xsl:with-param name="string" select="local-name()"/>
        </xsl:call-template>
        <xsl:text>: </xsl:text>
      </d:emphasis>
      <d:itemizedlist>
        <xsl:for-each select="./*">
          <d:listitem>
            <d:para>
              <xsl:apply-templates select="." mode="ism:portionmark"/>
              <xsl:value-of select="text()"/>
            </d:para>
          </d:listitem>
        </xsl:for-each>
      </d:itemizedlist>
    </d:para>
  </d:listitem>
</xsl:template>

<xsl:template match="boe:SSP/boe:Overlays" priority="5">
  <d:para>
    <d:emphasis role="bold">
      <xsl:apply-templates select="." mode="ism:portionmark"/>
      <xsl:text> Applicable Overlays: </xsl:text>
    </d:emphasis>
    <d:itemizedlist>
      <xsl:for-each select="./*">
        <d:listitem>
          <d:para>
            <xsl:value-of select="text()"/>
          </d:para>
        </d:listitem>
      </xsl:for-each>
    </d:itemizedlist>
  </d:para>
</xsl:template>

<xsl:template match="boe:SSP/boe:SecurityControlList" priority="5">
  <xsl:variable name="title">
    <xsl:call-template name="breakIntoWords">
```

```

    <xsl:with-param name="string" select="local-name()"/>
  </xsl:call-template>
</xsl:variable>
<d:para>
  <d:emphasis role="bold">
    <xsl:text>(U) </xsl:text>
    <xsl:value-of select="$title"/>
    <xsl:text>: </xsl:text>
  </d:emphasis>The following table is <xsl:apply-templates select="." mode="ism:banner"/>.</d:para>

<d:table frame="all">
  <d:title>(U) <xsl:value-of select="$title"/>
  </d:title>
  <d:tgroup cols="4">
    <d:colspec colname="c1" colnum="1" colwidth="1.0*" />
    <d:colspec colname="c2" colnum="2" colwidth="1.0*" />
    <d:colspec colname="c3" colnum="3" colwidth="1.0*" />
    <d:colspec colname="c4" colnum="4" colwidth="3.0*" />
    <d:thead>
      <d:row>
        <d:entry>Control Number</d:entry>
        <d:entry>Control Name</d:entry>
        <d:entry>Control Status</d:entry>
        <d:entry>Implementation Description</d:entry>
      </d:row>
    </d:thead>
    <d:tbody>
      <xsl:apply-templates select=".*" />
    </d:tbody>
  </d:tgroup>
</d:table>
</xsl:template>

<xsl:template match="boe:SecurityControl">
  <d:row>
    <d:entry>
      <xsl:value-of select="@boe:number"/>
    </d:entry>
    <d:entry>
      <xsl:value-of select="@boe:name"/>
    </d:entry>
    <d:entry>
      <xsl:value-of select="@boe:status"/>
    </d:entry>
    <d:entry>
      <xsl:if test="boe:ImplementationDescription">
        <d:para>
          <xsl:apply-templates select="boe:ImplementationDescription" mode="ism:portionmark"/>
          <xsl:value-of select="boe:ImplementationDescription"/>
        </d:para>
      </xsl:if>
      <xsl:if test="boe:Rationale">
        <d:para>
          <xsl:apply-templates select="boe:Rationale" mode="ism:portionmark"/> Rationale:

```

```

        <xsl:value-of select="boe:Rationale"/>
      </d:para>
    </xsl:if>
  </d:entry>
</d:row>
</xsl:template>

<xsl:template match="boe:Deficiency" priority="5">
  <xsl:apply-templates select="." mode="list"/>
</xsl:template>

<xsl:template match="boe:ThreatEvents" priority="5">

  <xsl:variable name="title">
    <xsl:call-template name="breakIntoWords">
      <xsl:with-param name="string" select="local-name()"/>
    </xsl:call-template>
  </xsl:variable>

  <d:para>
    <d:emphasis role="bold">
      <xsl:text>(U) </xsl:text>
      <xsl:value-of select="$title"/>
      <xsl:text>: </xsl:text>
    </d:emphasis>The following table is <xsl:apply-templates select="." mode="ism:banner"/>.</d:para>

    <d:table frame="all">
      <d:title>(U) <xsl:value-of select="$title"/>
      </d:title>
      <d:tgroup cols="7">
        <d:colspec colname="c1" colnum="1" colwidth="2.0*"/>
        <d:colspec colname="c2" colnum="2" colwidth="2.0*"/>
        <d:colspec colname="c3" colnum="3" colwidth="2.0*"/>
        <d:colspec colname="c4" colnum="4" colwidth="1.0*"/>
        <d:colspec colname="c5" colnum="5" colwidth="1.0*"/>
        <d:colspec colname="c6" colnum="6" colwidth="1.0*"/>
        <d:colspec colname="c7" colnum="7" colwidth="1.0*"/>
        <d:thead>
          <d:row>
            <d:entry>Event</d:entry>
            <d:entry>Vulnerabilities</d:entry>
            <d:entry>Sources</d:entry>
            <d:entry>Likelihood of Occurrence</d:entry>
            <d:entry>Likelihood of Success</d:entry>
            <d:entry>Overall Likelihood</d:entry>
            <d:entry>Residual Risk Level</d:entry>
          </d:row>
        </d:thead>
        <d:tbody>
          <xsl:apply-templates select="."/ *>
        </d:tbody>
      </d:tgroup>
    </d:table>
  </xsl:template>
```

```
<xsl:template match="boe:ThreatEvent">
  <d:row>
    <d:entry>
      <d:para>
        <xsl:apply-templates select="boe:Event"/>
      </d:para>
    </d:entry>
    <d:entry>
      <xsl:for-each select="boe:Vulnerabilities/boe:*">
        <d:para>
          <xsl:apply-templates select="."/>
        </d:para>
      </xsl:for-each>
    </d:entry>
    <d:entry>
      <xsl:for-each select="boe:ThreatSources/boe:*">
        <d:para>
          <xsl:apply-templates select="."/>
        </d:para>
      </xsl:for-each>
    </d:entry>
    <d:entry>
      <xsl:apply-templates select="boe:LikelihoodOfOccurance"/>
    </d:entry>
    <d:entry>
      <xsl:apply-templates select="boe:LikelihoodOfSuccess"/>
    </d:entry>
    <d:entry>
      <xsl:apply-templates select="boe:OverallLikelihood"/>
    </d:entry>
    <d:entry>
      <xsl:apply-templates select="boe:ResidualRiskLevel"/>
    </d:entry>
  </d:row>
</xsl:template>

<xsl:template match="boe:LikelihoodOfOccurance | boe:LikelihoodOfSuccess | boe:OverallLikelihood | boe:ResidualRiskLevel">
  <d:para>
    <xsl:apply-templates select="@boe:response"/>
  </d:para>
  <xsl:if test="text()">
    <d:para>
      <xsl:apply-templates select="text()"/>
    </d:para>
  </xsl:if>
</xsl:template>

<xsl:template match="boe:OverallRiskRatings" priority="5">
  <xsl:apply-templates select="." mode="list"/>
</xsl:template>

<xsl:template match="boe:SSP/*[matches(local-name(), '.*Diagram$')]" priority="4">
  <xsl:variable name="name" select="local-name()"/>
```

```

    <xsl:variable name="title">
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="$name"/>
      </xsl:call-template>
    </xsl:variable>
    <d:para>
      <d:emphasis role="bold">(U) <xsl:value-of select="$title"/>: </d:emphasis> The following diagram is the <xsl:value-of select="$title"/>
and is <xsl:apply-templates select="." mode="ism:banner"/>.</d:para>
    <d:mediaobject xml:id="{concat('image_', $name)}">
      <d:imageobject>
        <d:imagedata fileref="{string(boe:URI)}" align="center"/>
      </d:imageobject>
      <xsl:if test="boe:Description">
        <d:caption>
          <d:para>
            <xsl:apply-templates select="boe:Description" mode="ism:portionmark"/>
            <xsl:value-of select="boe:Description"/>
          </d:para>
        </d:caption>
      </xsl:if>
    </d:mediaobject>
  </xsl:template>

  <xsl:template match="boe:SSP/*[matches(local-name(), '.*Inventory$')]" priority="4">
    <xsl:variable name="title">
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name="type" select="tokenize(normalize-space($title), ' ')[1]"/>
    <d:para>
      <d:emphasis role="bold">
        <xsl:apply-templates select="." mode="ism:portionmark"/>
        <xsl:value-of select="$title"/>
        <xsl:text>: </xsl:text>
      </d:emphasis>
      <d:itemizedlist>
        <xsl:for-each select=".*">
          <d:listitem>
            <d:para>
              <d:emphasis role="bold">
                <xsl:apply-templates select="." mode="ism:portionmark"/>
                <xsl:value-of select="boe:Name"/>
              </d:emphasis>
            </d:para>
            <d:itemizedlist>
              <xsl:for-each select=".* except boe:Name">
                <d:listitem>
                  <xsl:variable name="name">
                    <xsl:choose>
                      <xsl:when test="$type='Hardware'">
                        <xsl:choose>
                          <xsl:when test="local-name()='Version'">
                            <xsl:value-of select="'Model Number'"/>
                          </xsl:when>

```



```

        <xsl:when test="local-name()='OwnerManufacturer'">
            <xsl:value-of select="'Manufacturer'"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()"/>
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>
</xsl:when>
<xsl:when test="$type='Software'">
    <xsl:choose>
        <xsl:when test="local-name()='OwnerManufacturer'">
            <xsl:value-of select="'Owner'"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name="breakIntoWords">
                <xsl:with-param name="string" select="local-name()"/>
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>
</xsl:when>
<xsl:otherwise>
    <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()"/>
    </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:call-template name="responseWithExplanation">
    <xsl:with-param name="title" select="$name"/>
    <xsl:with-param name="response" select="text()"/>
    <xsl:with-param name="hasExplanation" select="false()"/>
</xsl:call-template>
</d:listitem>
</xsl:for-each>
</d:itemizedlist>
</d:para>
</d:listitem>
</xsl:for-each>
</d:itemizedlist>
</d:para>
</xsl:template>

```

```

<!-- Person Types -->
<xsl:template match="boe:RiskAssessmentPOC | boe:AuthorizingOfficial | boe:PointOfContact | boe:SecurityAssessor"
    priority="5">
    <xsl:choose>
        <xsl:when test="local-name()='RiskAssessmentPOC'">
            <xsl:apply-templates select="." mode="list">
                <xsl:with-param name="title">
                    <xsl:text>Risk Assessment POC: </xsl:text>
                </xsl:with-param>
            </xsl:apply-templates>
        </xsl:when>
    </xsl:choose>

```

```

        </xsl:when>
        <xsl:otherwise>
            <xsl:apply-templates select="." mode="list"/>
        </xsl:otherwise>
    </xsl:choose>

</xsl:template>

<xsl:template match="boe:ConstraintsAndIssues" priority="5">
    <xsl:apply-templates select="." mode="list"/>
</xsl:template>

<xsl:template match="boe:ComponentsAssessed" priority="5">
    <xsl:apply-templates select="." mode="list"/>
</xsl:template>

<xsl:template match="boe:IdentifyingEvent" priority="5">
    <xsl:apply-templates select="." mode="list"/>
</xsl:template>

<xsl:template match="boe:SecurityControlTraceabilityList" priority="5">
    <xsl:variable name="title">
        <xsl:call-template name="breakIntoWords">
            <xsl:with-param name="string" select="local-name()"/>
        </xsl:call-template>
    </xsl:variable>
    <d:para>
        <d:emphasis role="bold">
            <xsl:text>(U) </xsl:text>
            <xsl:value-of select="$title"/>
            <xsl:text>: </xsl:text>
        </d:emphasis>The following table is <xsl:apply-templates select="." mode="ism:banner"/>.</d:para>

        <d:table frame="all">
            <d:title>(U) <xsl:value-of select="$title"/>
            </d:title>
            <d:tgroup cols="2">
                <d:colspec colname="c1" colnum="1" colwidth="1.0*" />
                <d:colspec colname="c2" colnum="2" colwidth="1.0*" />
                <d:thead>
                    <d:row>
                        <d:entry>Control Number</d:entry>
                        <d:entry>Requirement ID</d:entry>
                    </d:row>
                </d:thead>
                <d:tbody>
                    <xsl:apply-templates select="."/ *>
                </d:tbody>
            </d:tgroup>
        </d:table>
    </xsl:template>

<xsl:template match="boe:SecurityControlTrace">
    <d:row>
```

```

    <d:entry>
      <xsl:value-of select="@boe:controlNumber"/>
    </d:entry>
  </d:entry>
  <xsl:value-of select="@boe:requirementId"/>
</d:entry>
</d:row>
</xsl:template>

<xsl:template match="boe:TestResults" priority="5">
  <xsl:variable name="title">
    <xsl:call-template name="breakIntoWords">
      <xsl:with-param name="string" select="local-name()"/>
    </xsl:call-template>
  </xsl:variable>
  <d:para>
    <d:emphasis role="bold">
      <xsl:text>(U) </xsl:text>
      <xsl:value-of select="$title"/>
      <xsl:text>: </xsl:text>
    </d:emphasis>The following table is <xsl:apply-templates select="." mode="ism:banner"/>.</d:para>

    <d:table frame="all">
      <d:title>(U) <xsl:value-of select="$title"/>
    </d:title>
    <d:tgroup cols="5">
      <d:colspec colname="c1" colnum="1" colwidth="1.0*"/>
      <d:colspec colname="c2" colnum="2" colwidth="1.0*"/>
      <d:colspec colname="c3" colnum="3" colwidth="1.0*"/>
      <d:colspec colname="c4" colnum="4" colwidth="1.0*"/>
      <d:colspec colname="c5" colnum="5" colwidth="1.0*"/>
      <d:thead>
        <d:row>
          <d:entry>Test ID</d:entry>
          <d:entry>Test Date</d:entry>
          <d:entry>Resource Tested</d:entry>
          <d:entry>Control Number Tested</d:entry>
          <d:entry>Test Result</d:entry>
        </d:row>
      </d:thead>
      <d:tbody>
        <xsl:apply-templates select=".*"/>
      </d:tbody>
    </d:tgroup>
  </d:table>
</xsl:template>

<xsl:template match="boe:Test">
  <d:row>
    <d:entry>
      <xsl:apply-templates select="@boe:testId"/>
    </d:entry>
    <d:entry>
      <xsl:apply-templates select="@boe:date"/>
    </d:entry>
  </d:row>
</xsl:template>
```

```

        </d:entry>
        <d:entry>
            <xsl:apply-templates select="@boe:resource"/>
        </d:entry>
        <d:entry>
            <xsl:apply-templates select="@boe:controlNumber"/>
        </d:entry>
        <d:entry>
            <xsl:apply-templates select="text()"/>
        </d:entry>
    </d:row>
</xsl:template>

<!-- Handle children of main doc elements -->
<xsl:template match="boe:SystemInformation/* | boe:SSP/* | boe:SAR/* | boe:RAR/* | boe:POAM/* | boe:ADD/*">
    <xsl:choose>
        <xsl:when test="descendant::*">
            <xsl:apply-templates select="."/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:apply-templates select="." mode="list"/>
            <!--<xsl:variable name="responseText">
        <xsl:choose>
            <xsl:when test="@boe:response">
                <xsl:call-template name="capitalizeFirstLetter">
                    <xsl:with-param name="string" select="@boe:response"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="normalize-space(text())"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:variable name="explanationExists">
        <xsl:choose>
            <xsl:when test="@boe:response and (string-length(normalize-space(text())) > 0)">
                <xsl:value-of select="true()"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="false()"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:call-template name="responseWithExplanation">
        <xsl:with-param name="response" select="$responseText"/>
        <xsl:with-param name="hasExplanation" select="$explanationExists"/>
    </xsl:call-template>-->
</xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template match="boe:*[descendant::boe:*)" mode="list">
    <xsl:param name="title" as="xs:string">
    <xsl:variable name="words">

```

```

        <xsl:call-template name="breakIntoWords">
            <xsl:with-param name="string" select="local-name()"/>
        </xsl:call-template>
    </xsl:variable>
    <xsl:value-of select="concat(string-join($words,' '),': ')" />
</xsl:param>
<d:para>
    <d:emphasis role="bold">
        <xsl:apply-templates select="." mode="ism:portionmark"/>
        <xsl:value-of select="$title"/>
    </d:emphasis>
    <d:itemizedlist>
        <xsl:for-each select="boe:*">
            <d:listitem>
                <xsl:apply-templates select="." mode="list"/>
            </d:listitem>
        </xsl:for-each>
    </d:itemizedlist>

</d:para>
</xsl:template>

<xsl:template match="boe:*[not(descendant::boe:*)]" mode="list">
    <xsl:variable name="responseText">
        <xsl:choose>
            <xsl:when test="@boe:response">
                <xsl:call-template name="capitalizeFirstLetter">
                    <xsl:with-param name="string" select="@boe:response"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="normalize-space(text())"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:variable name="explanationExists">
        <xsl:choose>
            <xsl:when test="@boe:response and (string-length(normalize-space(text())) > 0)">
                <xsl:value-of select="true()"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="false()"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:call-template name="responseWithExplanation">
        <xsl:with-param name="response" select="$responseText"/>
        <xsl:with-param name="hasExplanation" select="$explanationExists"/>
    </xsl:call-template>
</xsl:template>

<!-- ===== -->
<!-- Named Templates -->
<!-- ===== -->

```

```

<xsl:template name="responseWithExplanation">
  <xsl:param name="title" as="xs:string">
    <xsl:variable name="wordList">
      <xsl:call-template name="breakIntoWords">
        <xsl:with-param name="string" select="local-name()"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:value-of select="string-join($wordList, ' ')" />
  </xsl:param>
  <xsl:param name="response" as="xs:string">
    <xsl:call-template name="capitalizeFirstLetter">
      <xsl:with-param name="string" select="@boe:response"/>
    </xsl:call-template>
  </xsl:param>
  <xsl:param name="hasExplanation" as="xs:boolean" select="not(empty(text()))"/>

  <d:para>
    <d:emphasis role="bold">
      <xsl:if test="@ism:*">
        <xsl:apply-templates select="." mode="ism:portionmark"/>
      </xsl:if>
      <xsl:value-of select="$title"/>: </d:emphasis>
      <xsl:value-of select="$response"/>
      <xsl:if test="$hasExplanation">
        <xsl:text> -- </xsl:text>
        <xsl:value-of select="text()"/>
      </xsl:if>
    </d:para>
  </xsl:template>

  <xsl:template name="breakIntoWords">
    <xsl:param name="string"/>
    <xsl:choose>
      <xsl:when test="string-length($string) < 2">
        <xsl:value-of select="$string"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name="breakIntoWordsHelper">
          <xsl:with-param name="string" select="$string"/>
          <xsl:with-param name="token" select="substring($string, 1, 1)" />
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>

  <xsl:template name="breakIntoWordsHelper">
    <xsl:param name="string" select="''"/>
    <xsl:param name="token" select="''"/>
    <xsl:choose>
      <xsl:when test="string-length($string) = 0"/>
      <xsl:when test="string-length($token) = 0"/>
      <xsl:when test="string-length($string) = string-length($token)">
        <xsl:value-of select="$token"/>
      </xsl:when>
    </xsl:choose>
  </xsl:template>

```

```

    </xsl:when>
    <xsl:when test="contains('ABCDEFGHIJKLMNOPQRSTUVWXYZ',substring($string, string-length($token) + 1, 1))">
      <xsl:value-of select="concat($token, ' ')" />
      <xsl:call-template name="breakIntoWordsHelper">
        <xsl:with-param name="string" select="substring-after($string, $token)" />
        <xsl:with-param name="token" select="substring($string, string-length($token), 1)" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="breakIntoWordsHelper">
        <xsl:with-param name="string" select="$string" />
        <xsl:with-param name="token" select="substring($string, 1, string-length($token) + 1)" />
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name="capitalizeFirstLetter">
  <xsl:param name="string" />
  <xsl:value-of select="concat(translate(substring($string, 1, 1),
substring($string,2,string-length($string)-1)
                                'abcdefghijklmnopqrstuvwxyz',
                                'ABCDEFGHIJKLMNOPQRSTUVWXYZ'),
                                )" />
</xsl:template>
</xsl:stylesheet>
```