



Guide to Schematron Rules and Patterns

PUBS Schematron Guide

Version 2016-SEP

September 9, 2016

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
1.2 - Overview	1
1.3 - Schematron	1
1.4 - Conformance	1
Chapter 2 - Rules	2
2.1 - ../Rules/PUBS_ID_00001.sch	3
2.2 - ../Rules/PUBS_ID_00003.sch	10
2.3 - ../Rules/PUBS_ID_00008.sch	13
2.4 - ../Rules/PUBS_ID_00009.sch	14
2.5 - ../Rules/PUBS_ID_00017.sch	15
2.6 - ../Rules/PUBS_ID_00027.sch	16
2.7 - ../Rules/PUBS_ID_00028.sch	17
2.8 - ../Rules/PUBS_ID_00029.sch	18
2.9 - ../Rules/PUBS_ID_00030.sch	19
2.10 - ../Rules/PUBS_ID_00031.sch	20
2.11 - ../Rules/PUBS_ID_00032.sch	21
2.12 - ../Rules/PUBS_ID_00033.sch	22
2.13 - ../Rules/PUBS_ID_00034.sch	23
2.14 - ../Rules/PUBS_ID_00035.sch	24
2.15 - ../Rules/PUBS_ID_00036.sch	25
2.16 - ../Rules/PUBS_ID_00040.sch	26
2.17 - ../Rules/PUBS_ID_00042.sch	28
2.18 - ../Rules/PUBS_ID_00044.sch	29
2.19 - ../Rules/PUBS_ID_00045.sch	30
2.20 - ../Rules/PUBS_ID_00046.sch	31
2.21 - ../Rules/PUBS_ID_00047.sch	32
2.22 - ../Rules/PUBS_ID_00048.sch	33
2.23 - ../Rules/PUBS_ID_00049.sch	34
2.24 - ../Rules/PUBS_ID_00050.sch	35
2.25 - ../Rules/PUBS_ID_00053.sch	36
2.26 - ../Rules/PUBS_ID_00054.sch	37
2.27 - ../Rules/PUBS_ID_00055.sch	40
2.28 - ../Rules/PUBS_ID_00062.sch	41
2.29 - ../Rules/PUBS_ID_00063.sch	42
2.30 - ../Rules/PUBS_ID_00064.sch	43
2.31 - ../Rules/PUBS_ID_00065.sch	44
2.32 - ../Rules/PUBS_ID_00070.sch	45
2.33 - ../Rules/PUBS_ID_00072.sch	46
2.34 - ../Rules/PUBS_ID_00079.sch	47
2.35 - ../Rules/PUBS_ID_00080.sch	48
2.36 - ../Rules/PUBS_ID_00081.sch	49
2.37 - ../Rules/PUBS_ID_00082.sch	50
2.38 - ../Rules/PUBS_ID_00086.sch	51
2.39 - ../Rules/PUBS_ID_00087.sch	52
2.40 - ../Rules/PUBS_ID_00092.sch	53

2.41 - ../Rules/PUBS_ID_00093.sch	54
2.42 - ../Rules/PUBS_ID_00096.sch	55
2.43 - ../Rules/PUBS_ID_00097.sch	56
2.44 - ../Rules/PUBS_ID_00098.sch	57
2.45 - ../Rules/PUBS_ID_00099.sch	58
2.46 - ../Rules/PUBS_ID_00100.sch	59
2.47 - ../Rules/PUBS_ID_00101.sch	60
2.48 - ../Rules/PUBS_ID_00103.sch	61
2.49 - ../Rules/PUBS_ID_00104.sch	62
2.50 - ../Rules/PUBS_ID_00110.sch	63
2.51 - ../Rules/PUBS_ID_00111.sch	64
2.52 - ../Rules/PUBS_ID_00113.sch	65
2.53 - ../Rules/PUBS_ID_00114.sch	66
2.54 - ../Rules/PUBS_ID_00115.sch	67
2.55 - ../Rules/PUBS_ID_00116.sch	68
2.56 - ../Rules/PUBS_ID_00117.sch	69
2.57 - ../Rules/PUBS_ID_00118.sch	70
2.58 - ../Rules/PUBS_ID_00119.sch	71
2.59 - ../Rules/PUBS_ID_00120.sch	72
2.60 - ../Rules/PUBS_ID_00121.sch	73
2.61 - ../Rules/PUBS_ID_00122.sch	74
2.62 - ../Rules/PUBS_ID_00123.sch	75
2.63 - ../Rules/PUBS_ID_00124.sch	76
2.64 - ../Rules/PUBS_ID_00125.sch	77
2.65 - ../Rules/PUBS_ID_00126.sch	78
2.66 - ../Rules/PUBS_ID_00127.sch	79
2.67 - ../Rules/PUBS_ID_00128.sch	80
2.68 - ../Rules/PUBS_ID_00129.sch	81
Chapter 3 - Abstract Patterns	82
3.1 - ../Lib/CompareDateTimes.sch	83
3.2 - ../Lib/ValidateValueExistenceInList.sch	84
Chapter 4 - Schematron Schema	85
4.1 - ../PUBS_XML.sch	86
Chapter 5 - Removed Rules	97
5.1 - ../Rules/deleted/PUBS_ID_00002.sch	97
5.2 - ../Rules/deleted/PUBS_ID_00004.sch	97
5.3 - ../Rules/deleted/PUBS_ID_00005.sch	97
5.4 - ../Rules/deleted/PUBS_ID_00006.sch	97
5.5 - ../Rules/deleted/PUBS_ID_00007.sch	97
5.6 - ../Rules/deleted/PUBS_ID_00010.sch	97
5.7 - ../Rules/deleted/PUBS_ID_00011.sch	97
5.8 - ../Rules/deleted/PUBS_ID_00012.sch	98
5.9 - ../Rules/deleted/PUBS_ID_00013.sch	98
5.10 - ../Rules/deleted/PUBS_ID_00014.sch	98
5.11 - ../Rules/deleted/PUBS_ID_00015.sch	98
5.12 - ../Rules/deleted/PUBS_ID_00016.sch	98
5.13 - ../Rules/deleted/PUBS_ID_00018.sch	98
5.14 - ../Rules/deleted/PUBS_ID_00019.sch	98
5.15 - ../Rules/deleted/PUBS_ID_00020.sch	99

5.16 - //Rules/deleted/PUBS_ID_00021.sch	99
5.17 - //Rules/deleted/PUBS_ID_00022.sch	99
5.18 - //Rules/deleted/PUBS_ID_00023.sch	99
5.19 - //Rules/deleted/PUBS_ID_00024.sch	99
5.20 - //Rules/deleted/PUBS_ID_00025.sch	99
5.21 - //Rules/deleted/PUBS_ID_00026.sch	99
5.22 - //Rules/deleted/PUBS_ID_00037.sch	100
5.23 - //Rules/deleted/PUBS_ID_00038.sch	100
5.24 - //Rules/deleted/PUBS_ID_00039.sch	100
5.25 - //Rules/deleted/PUBS_ID_00041.sch	100
5.26 - //Rules/deleted/PUBS_ID_00043.sch	100
5.27 - //Rules/deleted/PUBS_ID_00051.sch	100
5.28 - //Rules/deleted/PUBS_ID_00052.sch	100
5.29 - //Rules/deleted/PUBS_ID_00056.sch	101
5.30 - //Rules/deleted/PUBS_ID_00057.sch	101
5.31 - //Rules/deleted/PUBS_ID_00058.sch	101
5.32 - //Rules/deleted/PUBS_ID_00059.sch	101
5.33 - //Rules/deleted/PUBS_ID_00060.sch	101
5.34 - //Rules/deleted/PUBS_ID_00061.sch	101
5.35 - //Rules/deleted/PUBS_ID_00066.sch	101
5.36 - //Rules/deleted/PUBS_ID_00067.sch	102
5.37 - //Rules/deleted/PUBS_ID_00068.sch	102
5.38 - //Rules/deleted/PUBS_ID_00069.sch	102
5.39 - //Rules/deleted/PUBS_ID_00071.sch	102
5.40 - //Rules/deleted/PUBS_ID_00073.sch	102
5.41 - //Rules/deleted/PUBS_ID_00074.sch	102
5.42 - //Rules/deleted/PUBS_ID_00075.sch	102
5.43 - //Rules/deleted/PUBS_ID_00076.sch	103
5.44 - //Rules/deleted/PUBS_ID_00077.sch	103
5.45 - //Rules/deleted/PUBS_ID_00078.sch	103
5.46 - //Rules/deleted/PUBS_ID_00083.sch	103
5.47 - //Rules/deleted/PUBS_ID_00084.sch	103
5.48 - //Rules/deleted/PUBS_ID_00085.sch	103
5.49 - //Rules/deleted/PUBS_ID_00088.sch	103
5.50 - //Rules/deleted/PUBS_ID_00089.sch	104
5.51 - //Rules/deleted/PUBS_ID_00090.sch	104
5.52 - //Rules/deleted/PUBS_ID_00091.sch	104
5.53 - //Rules/deleted/PUBS_ID_00094.sch	104
5.54 - //Rules/deleted/PUBS_ID_00102.sch	104
5.55 - //Rules/deleted/PUBS_ID_00105.sch	104
5.56 - //Rules/deleted/PUBS_ID_00106.sch	104
5.57 - //Rules/deleted/PUBS_ID_00107.sch	105
5.58 - //Rules/deleted/PUBS_ID_00108.sch	105
5.59 - //Rules/deleted/PUBS_ID_00109.sch	105
5.60 - //Rules/deleted/PUBS_ID_00112.sch	105

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for PUBS. This guide is generated from the PUBS Schematron rules and provides a consolidated reference for the business rules of this specification.

1.2 - Overview

Chapter 2 is a listing of all the numbered rules in PUBS. For each rule, there is a rule description, a code description, and a code block with the Schematron rule.

Chapter 3 is a listing of abstract patterns used in PUBS. The abstract patterns may be used in numbered rules or provided as reference for use in rules developed by users of PUBS. Each abstract pattern has a code description and a code block with the abstract Schematron pattern.

Chapter 4 is a listing of the master ISM Schematron file with all of the imports of rules and patterns. Many of the rules and patterns listed in Chapters 3 and 4 rely on functions and variables defined in the master file.

Chapter 5 is a listing of rules that have been deleted.

1.3 - Schematron

The business rules for PUBS are encoded using ISO Schematron. Schematron is a rule-based validation language that uses XML Path Language to make assertions about an XML document.

PUBS uses the XSLT 2.0 implementation of Schematron by Rick Jelliffe (2010-04-14) as its reference implementation. The only available identifying descriptors for this implementation are the implementer's name and date of release. This implementation may be found at the following URL: <http://code.google.com/p/schematron/>.



Important

The Schematron rules in this specification use XSLT 2.0 query binding.

1.4 - Conformance

This guide is informative. The Schematron rules listed here are normative in the sense that they convey criteria that a document **MUST** adhere to, exactly as English may be used to convey normative criteria. It is not necessary for implementers to use the specific Schematron encoding in this specification. Implementers **MAY** use any encodings, tools, or languages desired to implement validation schemes for conformance to this specification. However, to conform to the specification, validation schemes **MUST** match the behavior of the reference Schematron implementation. That is, a validator **MUST** find a document valid *if and only if* the reference Schematron implementation would find the document valid according to PUBS's Schematron rules.

Chapter 2 - Rules

All of the numbered Rules for PUBS are listed in this section. These rules may depend on patterns defined in the Abstract Patterns section or on variables defined in the Schematron Schema section.

Rules identifiers are all of the format PUBS-ID-XXXXX, with rule files named PUBS_ID_XXXXX.sch. Any other heading indicates a supporting file that may influence a rule but is not actually a numbered rule.

2.1 - `./Rules/PUBS_ID_00001.sch`

Rule Description

[PUBS-ID-00001][Warning] For every optional element that exists in the document and can have text content, the element should have non-null, non-whitespace value.

Code Description

This pattern uses an abstract rule to consolidate logic. The abstract rule first concatenates the text values within the given element, separated by a single space. The resultant string is then normalized with leading and trailing whitespace removed, and the length of the string is determined to be greater than zero, which indicates non-whitespace content. The abstract rule is extended once for each optional element in the PUBS schema.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00001">
    <sch:rule abstract="true" id="abs.rule00001">
        <sch:assert test="normalize-space(string())" flag="warning">[PUBS-ID-00001][Warning] For every optional element that exists in the document and can have text content,
the element should have non-null, non-whitespace value.</sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule on optional elements --><!-- mixed='true' -->
<sch:rule context="pubs:AbbreviatedTitle">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:AttributionText">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:BiographicalSketch">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Caption">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:CompilationTitle">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Description">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Drug">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:EmphasizedText">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Footnote">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Legend">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Link">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:ListItem">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Para">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:PreformattedText">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
```

```
<sch:rule context="pubs:SourceDescriptor">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:SourcedText">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Subtitle">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:TextBox">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Title">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:entry">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<!-- Extensions of RunningTextType -->
<sch:rule context="pubs:Account">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:CommData">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Commodity">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Concept">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:CountryName">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Date">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:DateTime">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:EntityUntyped">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Equipment">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Event">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Facility">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:GeoFeature">
  <sch:extends rule="abs.rule00001"/>
```

```
</sch:rule>
<sch:rule context="pubs:GeoRef">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Identifier">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:InfoBearer">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:LocationOfInterest">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:MilitaryUnit">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Money">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Nomenclature">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:NoteInline">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Organization">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Person">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:QuantityReference">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Quote">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:SystemClass">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Term">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Time">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Vehicle">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Weapon">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<!-- Things with Strings -->
<sch:rule context="pubs:Abbreviation">
```

```
<sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Acronym">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:AddressLine">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:BibliographyEntry">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:City">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:ContactFor">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:CopyrightAttribution">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:DocumentID">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:EditionNumber">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:GenerationalQualifier">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:GivenName">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:GlossaryTerm">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Honorific">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:IndexEntry">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:IssueNumber">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:JobTitle">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:MediaExtent">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:MilitaryRank">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:OfficeName">
```

```
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:OriginalClassificationMarking">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:PersonalTitle">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:PostalCode">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:Province">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:SegmentLabel">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:SourceID">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:State">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:Subscript">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:Superscript">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:Surname">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:UUID">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:VolumeNumber">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <sch:rule context="pubs:locatorLink">
        <sch:extends rule="abs.rule00001"/>
    </sch:rule>
    <!-- extension of xsd:string -->
<sch:rule context="pubs:Affiliation">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:AlternateFormat">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:CountryCode">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:EmailAddress">
    <sch:extends rule="abs.rule00001"/>
</sch:rule>
```

```
<sch:rule context="pubs:FaxNumber">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:FormattedSignatureBlock">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:InternalID">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:NetworkAddress">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:OtherProperty">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:PhoneNumber">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:PlaceName">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:ProductLine">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:Region">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:SubjectCategory">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:SubjectCode">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:UserID">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<sch:rule context="pubs:WebPageAddress">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
<!-- IRM Related elements -->
<sch:rule context="ddms:NonStateActor">
  <sch:extends rule="abs.rule00001"/>
</sch:rule>
</sch:pattern>
```

2.2 - ./Rules/PUBS_ID_00003.sch

Rule Description

[PUBS-ID-00003][Error] The following elements, if they exist in an XML instance, must have content: ddms:taskID, DescriptiveMetadata/Title, DescriptiveMetadata/Description, ddms:mimeType, ddms:applicationSoftware, NetworkAddress, Appendix/Title, BibliographyDivision/Title, BibliographyEntry, Definition/Para, DistributionEntry, GlossaryDivision/Title, GlossaryTerm, IndexDivision/Title, IndexEntry, ListItem, LongDescription/Para, Section/Title, DateString. Human Readable: All of the following elements must have a value if they exist in an XML instance: ddms:taskID, DescriptiveMetadata/Title, DescriptiveMetadata/Description, ddms:mimeType, ddms:applicationSoftware, NetworkAddress, Appendix/Title, BibliographyDivision/Title, BibliographyEntry, Definition/Para, DistributionEntry, GlossaryDivision/Title, GlossaryTerm, IndexDivision/Title, IndexEntry, ListItem, LongDescription/Para, Section/Title, DateString.

Code Description

This pattern uses an abstract rule to consolidate logic. The abstract rule normalizes the space of the value of the content element and makes sure that the length of the resulting string is greater than zero, which indicates non-whitespace content. The abstract rule is extended once for each element included in rule PUBS_ID_00003.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00003"><!-- Abstract rule, which asserts that the context element has content -->
<sch:rule abstract="true" id="abs.rule00003">
    <sch:assert test="normalize-space(string(.))" flag="error">[PUBS-ID-00003][Error] The element
    <sch:name/>must have content. Human Readable: All of the following elements must have a value if they exist in an XML instance: ddms:taskID, DescriptiveMetadata/Title,
DescriptiveMetadata/Description, ddms:mimeType, ddms:applicationSoftware, NetworkAddress, Appendix/Title, BibliographyDivision/Title, BibliographyEntry, Definition/Para, DistributionEntry,
GlossaryDivision/Title, GlossaryTerm, IndexDivision/Title, IndexEntry, ListItem, LongDescription/Para, Section/Title, DateString.
    </sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:taskID">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:DescriptiveMetadata/pubs:Title">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:DescriptiveMetadata/pubs:Description">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:mimeType">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:applicationSoftware">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:NetworkAddress">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:Appendix/pubs:Title">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:BibliographyDivision/pubs:Title">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:BibliographyEntry">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:Definition/pubs:Para">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:DistributionEntry">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:GlossaryDivision/pubs:Title">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:GlossaryTerm">
    <sch:extends rule="abs.rule00003"/>
</sch:rule>
```



```
<sch:rule context="pubs:IndexDivision/pubs:Title">
  <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:IndexEntry">
  <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:ListItem">
  <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:LongDescription/pubs:Para">
  <sch:extends rule="abs.rule00003"/>
</sch:rule>
<sch:rule context="pubs:Section/pubs:Title">
  <sch:extends rule="abs.rule00003"/>
</sch:rule>
</sch:pattern>
```

2.3 - ../Rules/PUBS_ID_00008.sch

Rule Description

[PUBS-ID-00008][Error] The IntelDoc element must have element content in at least one of the descendent elements. Human Readable: Element IntelDoc must have content in at least one child element.

Code Description

This calculates the string value of the pubs:IntelDoc element if the value contains any non white space, then the rule returns true.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00008">
    <sch:rule context="pubs:DocumentBody">
        <sch:assert test="string-length(normalize-space(string-join(descendant-or-self::text(),'')))>0"
            flag="error">[PUBS-ID-00008][Error] The IntelDoc element must have element content in at least one of the descendent elements. Human Readable: Element
IntelDoc must have content in at least one child element.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.4 - ../Rules/PUBS_ID_00009.sch

Rule Description

[PUBS-ID-00009][Error] For element SourceReference, either child element SourceID or child element DocumentID must exist and must have a non-null value.

Code Description

If element SourceID is specified and has non-white space content or is element DocumentID is specified and has non-whitespace content, then the rule returns true.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00009">
    <sch:rule context="pubs:SourceReference">
        <sch:assert test="normalize-space(string(pubs:SourceID)) or normalize-space(string(pubs:DocumentID))"
            flag="error">[PUBS-ID-00009][Error] For element SourceReference, either child element SourceID or child element DocumentID must exist and must have a non-
null value.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.5 - ../Rules/PUBS_ID_00017.sch

Rule Description

[PUBS-ID-00017][Error] If element CountryName has either attribute countryCode or countryCodeVocabulary is specified, then both must be specified. Human Readable: Element CountryName must have a value for both the countryCode and countryCodeVocabulary attributes.

Code Description

This rule ensures that element CountryName has either both attribute countryCode and attribute countryCodeVocabulary specified, or that element CountryName has neither attribute countryCode nor attribute countryCodeVocabulary specified.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00017">
    <sch:rule context="pubs:CountryName">
        <sch:assert test="(@countryCode and @countryCodeVocabulary) or not(@countryCode or @countryCodeVocabulary)"
            flag="error">[PUBS-ID-00017][Error] If element CountryName has either attribute countryCode or countryCodeVocabulary is specified, then both must be
specified. Human Readable: Element CountryName must have a value for both the countryCode and countryCodeVocabulary attributes.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.6 - ../Rules/PUBS_ID_00027.sch

Rule Description

[PUBS-ID-00027][Error] For element ImageArea attribute coordinates must be specified and the value must be appropriate to the value of attribute shape as defined by the PUBS.XML Data Element Dictionary. Human Readable: Element ImageArea attribute coordinates and shape must be specified. The value for attribute coordinates has to follow the format for the designated shape as defined by the PUBS.XML Data Element Dictionary.

Code Description

If attribute coordinates or attribute shape are not specified or are specified with an empty value, then return false. If the attribute shape is specified, then this rule uses a regular expression to make sure that the attribute coordinates is specified with an appropriate value from the PUBS.XML Data Element Dictionary, reproduced below for convenience: The number and order of values depends on the shape being defined. Possible combinations: - rect: left-x, top-y, right-x, bottom-y. - circle: center-x, center-y, radius. Note: When the radius value is a percentage value, user agents should calculate the final radius value based on the associated object's width and height. The radius should be the smaller value of the two. - poly: x1, y1, x2, y2, ..., xN, yN. The first x and y coordinate pair and the last should be the same to close the polygon. When these coordinate values are not the same, user agents should infer an additional coordinate pair to close the polygon. Coordinates are relative to the top, left corner of the object. All values are lengths. All values are separated by commas.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00027">
  <sch:rule context="pubs:ImageArea">
    <sch:let name="coordinatesAttr" value="normalize-space(string(@coordinates))"/>
    <sch:let name="shapeAttr" value="normalize-space(string(@shape))"/>
    <sch:let name="rectRegEx" value="'(-?[0-9]+,){3}-?[0-9]+$'"/>
    <sch:let name="circCoordinatesRegEx" value="'^(-?[0-9]+,){2}-?[0-9]+$'"/>
    <sch:let name="circPercentageRegEx"
      value="'^(-?[0-9]+,){2}(0|[1-9][1-9]?|100)%$'"/>
    <sch:let name="polyRegEx" value="'^((-?[0-9]+,){2})*-?[0-9]+,-?[0-9]+$'"/>
    <sch:assert test="if(string-length($coordinatesAttr) = 0 or string-length($shapeAttr) = 0) then false() else if($shapeAttr='rect') then matches(string($coordinatesAttr),
$rectRegEx) else if($shapeAttr='circ') then matches(string($coordinatesAttr),$circCoordinatesRegEx) or matches(string($coordinatesAttr),$circPercentageRegEx) else if($shapeAttr='poly') then
matches(string($coordinatesAttr),$polyRegEx) else false()"
      flag="error">[PUBS-ID-00027][Error] For element ImageArea attribute coordinates must be specified and the value must be appropriate to the value of
attribute shape as defined by the PUBS.XML Data Element Dictionary. Human Readable: Element ImageArea attribute coordinates and shape must be specified. The value for attribute coordinates has
to follow the format for the designated shape as defined by the PUBS.XML Data Element Dictionary.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.7 - ../Rules/PUBS_ID_00028.sch

Rule Description

[PUBS-ID-00028][Error] For elements AnimationExhibit, AudioExhibit, Interactive3DExhibit, OtherExhibit, StillImageExhibit and VideoExhibit attribute xlink:href must be specified. Human Readable: Attribute xlink:href must be specified for elements AnimationExhibit, AudioExhibit, Interactive3DExhibit, OtherExhibit, StillImageExhibit and VideoExhibit.

Code Description

This pattern uses an abstract rule to consolidate logic. It makes sure that the attribute xlink:href is specified. The abstract rule is extended once for each required element in rule PUBS_ID_00028.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00028"><!-- Abstract rule, which asserts that attribute xlink:href must be specified in the current context -->
<sch:rule abstract="true" id="abs.rule00028">
    <sch:assert test="@xlink:href" flag="error">[PUBS-ID-00028][Error] For element
    <sch:name/>attribute xlink:href must be specified.
  </sch:assert>
  </sch:rule>
  <!-- Begin using abstract rule to check required elements -->
<sch:rule context="pubs:AnimationExhibit">
  <sch:extends rule="abs.rule00028"/>
</sch:rule>
<sch:rule context="pubs:AudioExhibit">
  <sch:extends rule="abs.rule00028"/>
</sch:rule>
<sch:rule context="pubs:Interactive3DExhibit">
  <sch:extends rule="abs.rule00028"/>
</sch:rule>
<sch:rule context="pubs:OtherExhibit">
  <sch:extends rule="abs.rule00028"/>
</sch:rule>
<sch:rule context="pubs:StillImageExhibit">
  <sch:extends rule="abs.rule00028"/>
</sch:rule>
<sch:rule context="pubs:VideoExhibit">
  <sch:extends rule="abs.rule00028"/>
</sch:rule>
</sch:pattern>
```

2.8 - ../Rules/PUBS_ID_00029.sch

Rule Description

[PUBS-ID-00029][Error] For elements Facility and Person, if attribute xlink:href exists, then the attribute must have a non-null value. Human Readable: If attribute xlink:href exists for elements Facility and Person, it must have a value.

Code Description

This pattern uses an abstract rule to consolidate logic. It normalizes the space of the value of attribute xlink:href and makes sure the length of the resulting string is greater than zero, which indicates non-whitespace content. The abstract rule is extended once for each required element in rule PUBS_ID_00029.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00029"><!-- Abstract rule, which asserts that if attribute xlink:href exists, then it must have a non-null value -->
<sch:rule abstract="true" id="abs.rule00029">
    <sch:assert test="normalize-space(string(@xlink:href))" flag="error">[PUBS-ID-00029][Error] For element
    <sch:name/>if attribute xlink:href exists, then the attribute must have a non-null value.
  </sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="pubs:Facility[@xlink:href]">
    <sch:extends rule="abs.rule00029"/>
  </sch:rule>
<sch:rule context="pubs:Person[@xlink:href]">
    <sch:extends rule="abs.rule00029"/>
  </sch:rule>
</sch:pattern>
```

2.9 - ../Rules/PUBS_ID_00030.sch

Rule Description

[PUBS-ID-00030][Error] For element Glossary, if child element GlossaryDivision is used, there must be at least two instances of element GlossaryDivision. Human Readable: Element Glossary child element GlossaryDivision requires more than one instance.

Code Description

This rule ensures that element Glossary does not contain exactly one GlossaryDivision child element.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00030">
    <sch:rule context="pubs:Glossary">
        <sch:assert test="count(pubs:GlossaryDivision) != 1" flag="error">[PUBS-ID-00030][Error] For element Glossary, if child element GlossaryDivision is used, there must be
at least two instances of element GlossaryDivision. Human Readable: Element Glossary child element GlossaryDivision requires more than one instance.</sch:assert>
    </sch:rule>
</sch:pattern>
```


2.10 - ../Rules/PUBS_ID_00031.sch

Rule Description

[PUBS-ID-00031][Error] For element Bibliography, if child element BibliographyDivision is used, there must be at least two instances of element BibliographyDivision. Human Readable: Element Bibliography child element BibliographyDivision requires more than one instance.

Code Description

This rule ensures that element Bibliography does not contain exactly one BibliographyDivision child element.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00031">
    <sch:rule context="pubs:Bibliography">
        <sch:assert test="count(pubs:BibliographyDivision) != 1" flag="error">[PUBS-ID-00031][Error] For element Bibliography, if child element BibliographyDivision is used,
there must be at least two instances of element BibliographyDivision. Human Readable: Element Bibliography child element BibliographyDivision requires more than one instance.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.11 - ../Rules/PUBS_ID_00032.sch

Rule Description

[PUBS-ID-00032][Error] For element Index, if child element IndexDivision is used, there must be at least two instances of element IndexDivision. Human Readable: Element Index child element IndexDivision requires more than one instance.

Code Description

This rule ensures that element Index does not contain exactly one IndexDivision child element.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00032">
    <sch:rule context="pubs:Index">
        <sch:assert test="count(pubs:IndexDivision) != 1" flag="error">[PUBS-ID-00032][Error] For element Index, if child element IndexDivision is used, there must be at least
two instances of element IndexDivision. Human Readable: Element Index child element IndexDivision requires more than one instance.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.12 - ../Rules/PUBS_ID_00033.sch

Rule Description

[PUBS-ID-00033][Error] element ImageMap must specify attribute id. Human Readable: Element ImageMap must have a value for attribute id.

Code Description

Ensure the the id attribute is specified with non-whitespace characters

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00033">
    <sch:rule context="pubs:ImageMap">
        <sch:assert test="normalize-space(string(@id))" flag="error">[PUBS-ID-00033][Error] Element ImageMap must specify attribute id. Human Readable: Element ImageMap must
have a value for attribute id.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.13 - .//Rules/PUBS_ID_00034.sch

Rule Description

[PUBS-ID-00034][Error] For a given attribute imageMapReference value for element StillImageExhibit, there must be an attribute id for element ImageMap with an identical value within the XML instance. Human Readable: Element StillImageExhibit attribute imageMapReference value must equal element ImageMap attribute id value.

Code Description

If element StillImageExhibit has attribute imageMapReference specified, then this rule ensures that there exists in the document at least one ImageMap element with attribute id specified with a value equal to the value of attribute imageMapReference on element StillImageExhibit.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00034">
    <sch:rule context="pubs:StillImageExhibit[@imageMapReference]">
        <sch:let name="imageMapReference" value="@imageMapReference"/>
        <sch:assert test="ancestor::tdf:TrustedDataObject//pubs:ImageMap[@id=$imageMapReference]"
            flag="error">[PUBS-ID-00034][Error] For a given attribute imageMapReference value for element StillImageExhibit, there must be an attribute id for element
ImageMap with an identical value within the XML instance. Human Readable: Element StillImageExhibit attribute imageMapReference value must equal element ImageMap attribute id value.</
sch:assert>
    </sch:rule>
</sch:pattern>
```

2.14 - ../Rules/PUBS_ID_00035.sch

Rule Description

[PUBS-ID-00035][Error] All links that have attribute xlink:type specified with a value of [simple] must also have attribute xlink:href specified. Human Readable: If attribute xlink:type value is [simple], then attribute xlink:href has to have a value.

Code Description

For all elements that have the attribute xlink:type specified with a value of [simple], this rule normalizes the space of the value of the attribute xlink:href and makes sure that the length of the resulting string is greater than zero, which indicates non-whitespace content.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00035">
    <sch:rule context="pubs:*[@xlink:type='simple']">
        <sch:assert test="normalize-space(string(@xlink:href))" flag="error">[PUBS-ID-00035][Error] All links that have attribute xlink:type specified with a value of [simple]
must also have attribute xlink:href specified. Human Readable: If attribute xlink:type value is [simple], then attribute xlink:href has to have a value.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.15 - ../Rules/PUBS_ID_00036.sch

Rule Description

[PUBS-ID-00036][Error] Internal links must be traversable. If the attribute xlink:href value for elements Link and ImageArea is a URL fragment identifier with a [#] prefix, that fragment identifier must match the attribute id value of an element within the XML instance. Note that a fragment identifier is characterized by a [#] prefix. The [#] prefix is not part of the fragment identifier. So when the xlink:href attribute value starts with a [#], the remainder of the value is the fragment identifier. This rule does not apply to attribute values which contain a URL and a fragment identifier separated by [#], in which case it can not be determined with certainty that the target is within the XML instance. Human Readable: If the elements Link and ImageArea attribute xlink:href value starts with [#], then the fragment identifier portion of the value of attribute xlink:href must match the value of any attribute id specified within the XML instance.

Code Description

This pattern uses an abstract rule to consolidate logic. If the current element has attribute xlink:href specified with a value starting with [#], then this rule ensures that the fragment identifier portion of the value of attribute xlink:href matches the value of any attribute id specified within the XML instance. If the current element has attribute xlink:href specified with a value that does not start with [#], then this rule does not apply and this rule returns true.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00036"><!-- Abstract rule definition -->
<sch:rule abstract="true" id="abs.rule00036">
    <sch:let name="fragementIdentifier" value="substring(@xlink:href,2)"/>
    <sch:assert test="if(starts-with(@xlink:href,'#')) then ancestor::tdf:TrustedDataObject//*[@id=$fragementIdentifier] else true()"
        flag="error">[PUBS-ID-00036][Error] Internal links must be traversable. If the attribute xlink:href value for elements Link and ImageArea is a URL fragment
identifier with a [#] prefix, that fragment identifier must match the attribute id value of an element within the XML instance. Note that a fragment identifier is characterized by a [#]
prefix. The [#] prefix is not part of the fragment identifier. So when the xlink:href attribute value starts with a [#], the remainder of the value is the fragment identifier. This rule does
not apply to attribute values which contain a URL and a fragment identifier separated by [#], in which case it can not be determined with certainty that the target is within the XML instance.
Human Readable: If the elements Link and ImageArea attribute xlink:href value starts with [#], then the fragment identifier portion of the value of attribute xlink:href must match the value of
any attribute id specified within the XML instance.</sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="pubs:Link">
    <sch:extends rule="abs.rule00036"/>
</sch:rule>
<sch:rule context="pubs:ImageArea">
    <sch:extends rule="abs.rule00036"/>
</sch:rule>
</sch:pattern>
```

2.16 - `./Rules/PUBS_ID_00040.sch`

Rule Description

[PUBS-ID-00040][Error] The permissible values for the year range are 1901 through the current year for attributes ddns:approvedOn, ddns:posted, and ddms:receivedOn, and temporalCoverage elements with name element DatePublished, DateReviewed, and DateRevised. Human Readable: Year values for attributes ddns:approvedOn, ddns:posted, and ddms:receivedOn, and temporalCoverage elements with name element DatePublished, DateReviewed, and DateRevised have to fall within the range 1901 through the current year.

Code Description

This pattern uses abstract rules to consolidate logic. For elements, this rule ensures that the date contained within \$dateValue has a year value within the range \$minYear and \$maxYear, inclusive. For attributes, this rule ensures that each date contained within \$dateValues has a year value within the range \$minYear and \$maxYear, inclusive.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00040">
    <sch:let name="minYear" value="1901"/>
    <sch:let name="maxYear" value="$currentYear"/>
    <sch:rule abstract="true" id="abs.rule00040">
        <sch:let name="errMsg"
            value="concat('[PUBS-ID-00040][Error] The permissible values for the year range are 1901 through the current year for element ', name())"/>
        <sch:extends rule="abs.dateYearRangeRule"/>
    </sch:rule>
    <sch:rule abstract="true" id="abs.rule00040attrs">
        <sch:let name="fails"
            value="for $date in $dateValues return if(dtf:compareDateTimeRanges(string($minYear), '&lt;',' ', string($date)) and dtf:compareDateTimeRanges(string($date),
'&lt;',' ', string($maxYear))) then null else name($date)"/>
        <sch:assert test="count($fails)=0" flag="error">[PUBS-ID-00040][Error] The permissible values for the year range are 1901 through the current year for attribute(s):
        <sch:value-of select="for $each in $fails return concat(' ', string($each))"/>.
    </sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="pubs:AdministrativeMetadata/pubs:DateList/pubs:DatePublished">
    <sch:let name="dateValue" value="."/>
    <sch:extends rule="abs.rule00040"/>
</sch:rule>
    <!--<sch:rule context="ddms:temporalCoverage[ddms:name='DatePublished']/ddms:end">
    <sch:let name="dateValue" value="."/>
    <sch:extends rule="abs.rule00040"/>
</sch:rule>-->
<sch:rule context="pubs:AdministrativeMetadata/pubs:DateList/pubs:DateReviewed">
    <sch:let name="dateValue" value="."/>
    <sch:extends rule="abs.rule00040"/>
</sch:rule>
    <!--<sch:rule context="ddms:temporalCoverage[ddms:name='DateReviewed']/ddms:end">
    <sch:let name="dateValue" value="."/>
    <sch:extends rule="abs.rule00040"/>
</sch:rule>-->
<sch:rule context="pubs:AdministrativeMetadata/pubs:DateList/pubs:DateRevised">
    <sch:let name="dateValue" value="."/>
    <sch:extends rule="abs.rule00040"/>
</sch:rule>
    <!--<sch:rule context="ddms:temporalCoverage[ddms:name='DateRevised']/ddms:end">
    <sch:let name="dateValue" value="."/>
    <sch:extends rule="abs.rule00040"/>
</sch:rule>--></sch:pattern>
```


2.17 - ../Rules/PUBS_ID_00042.sch

Rule Description

[PUBS-ID-00042][Error] The permissible values for the year range are 0001 through 9999 for element DateTimeReferenced, and attributes date, dateTime, normalizedDate and normalizedDateTime. Human Readable: Year values for element DateTimeReferenced, and attributes date, dateTime, normalizedDate and normalizedDateTime have to fall within the range 0001 through 9999.

Code Description

This pattern uses abstract rules to consolidate logic. For elements, this rule ensures that the date contained within \$dateValue has a year value within the range \$minYear and \$maxYear, inclusive. For attributes, this rule ensures that each date contained within \$dateList has a year value within the range \$minYear and \$maxYear, inclusive.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00042"><!-- Use abstract rule to handle required elements -->
<sch:rule context="pubs:DateTimeReferenced">
    <sch:let name="minYear" value="0001"/>
    <sch:let name="maxYear" value="9999"/>
    <sch:let name="dateValue" value="."/>
    <sch:let name="errMsg"
        value="" [PUBS-ID-00042][Error] The permissible values for the year range are 0001 through 9999 for element DateTimeReferenced. ""/>
    <sch:extends rule="abs.dateYearRangeRule"/>
</sch:rule>
<!-- Use abstract rule to handle required attributes -->
<sch:rule context="pubs:*[@date] | pubs:*[@dateTime] | pubs:*[@normalizedDate] | pubs:*[@normalizedDateTime]">
    <sch:let name="minYear" value="0001"/>
    <sch:let name="maxYear" value="9999"/>
    <sch:let name="dateList"
        value="string-join((string(@date), string(@dateTime), string(@normalizeDate), string(@normalizedDateTime)), ' ')">
    <sch:let name="errMsg"
        value="" [PUBS-ID-00042][Error] The permissible values for the year range are 0001 through 9999 for attributes date, dateTime, normalizedDate, and
normalizedDateTime. ""/>
    <sch:extends rule="abs.dateListYearRangeRule"/>
</sch:rule>
</sch:pattern>
```

2.18 - ../Rules/PUBS_ID_00044.sch

Rule Description

[PUBS-ID-00044][Error] For attribute dateTimeRange, for each pair of date/time values, the second value must be later than the first value. Human Readable: The second value of date/time value pair for attribute dateTimeRange has to be later than the first value.

Code Description

The value of the attribute dateTimeRange is tokenized into a list of dateTimes, called \$dateTimeList. If the number of dates in \$dateTimeList is not even, then each date does not have a corresponding pair so this rule returns false. Otherwise, this rule verifies that the second date of each dateTime pair is later than the first date in that pair. To do this, the rule loops over all dates in \$dateTimeList and identify dateTime pairs by pairing the date at an even index N with the date at index N-1. For each pair, this rule verifies that \$dateList[N-1] is less than \$dateList[N].

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00044">
    <sch:rule context="pubs:*[@dateTimeRange]">
        <sch:let name="dateTimeList" value="tokenize(string(@dateTimeRange), ' ')" />
        <sch:assert test="if ((count($dateTimeList) mod 2) != 0) then false() else count( for $index in 1 to count($dateTimeList) return if($index mod 2 = 0) then
if(dtf:compareDateTimes($dateTimeList[$index - 1], '&lt;', $dateTimeList[$index])) then 1 else null else null ) = count($dateTimeList) * .5"
            flag="error">[PUBS-ID-00044][Error] For attribute dateTimeRange, for each pair of date/time values, the second value must be later than the first value.
Human Readable: The second value of date/time value pair for attribute dateTimeRange has to be later than the first value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.19 - ../Rules/PUBS_ID_00045.sch

Rule Description

[PUBS-ID-00045][Warning] DateApproved must be earlier than or equivalent to DatePublished and DatePosted. Human Readable: DateApproved has to be less than or equal to DatePublished and DatePosted.

Code Description

For each ddms:dates element which specifies attribute ddms:approvedOn, this rule ensures that the date specified in attribute ddms:approvedOn is earlier than or equivalent to the date specified in attribute ddms:posted and the date specified in the ddms:temporalCoverage element with ddms:name [DatePublished].

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00045">
  <sch:rule context="tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:dates[@ddms:approvedOn]">
    <sch:let name="secondaryDateList"
      value="(ancestor::tdf:TrustedDataObject//pubs:AdministrativeMetadata/pubs:DateList/pubs:DatePublished, @ddms:datePosted)"/>
    <sch:assert test="every $secondaryDate in $secondaryDateList satisfies dtf:compareDateTimeRanges(string(@ddms:approvedOn), '&lt;=', string($secondaryDate))"
      flag="warning">[PUBS-ID-00045][Warning][PUBS-ID-00045][Warning] DateApproved must be earlier than or equivalent to DatePublished and DatePosted. Human
Readable: DateApproved has to be less than or equal to DatePublished and DatePosted.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.20 - ../Rules/PUBS_ID_00046.sch

Rule Description

[PUBS-ID-00046][Error] DateInfoCutoff must be earlier than or equivalent to DateApproved, DatePublished and DatePosted. Human Readable: DateInfoCutoff has to be less than or equal to DateApproved, DatePublished, and DatePosted.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param \$context to each date contained within the param \$secondaryDateList (using the comparison operator contained in param \$operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be found in the abstract pattern definition file located in the Lib directory.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="CompareDateTimes" id="PUBS-ID-00046">
  <sch:param name="ruleText"
    value="' [PUBS-ID-00046][Error] DateInfoCutoff must be earlier than or equivalent to DateApproved, DatePublished and DatePosted. '"/>
  <sch:param name="codeDesc"
    value="' This rule uses an abstract pattern to consolidate logic. It compares the date contained within the
param $secondaryDateList (using the comparison operator contained in param $operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be
found in the abstract pattern definition file located in the Lib directory. '"/>
  <sch:param name="context"
    value="tdf:*[descendant::tdf:StructuredPayload/pubs:IntelDoc]//tdf:Assertion//ddms:temporalCoverage[ddms:name='infoCutOff']/ddms:approximableEnd/
ddms:searchableDate/ddms:end"/>
  <sch:param name="operator" value="'<='"/>
  <sch:param name="secondaryDateList"
    value="(ancestor::ddms:resource/ddms:dates/@ddms:approvedOn, ancestor::tdf:TrustedDataObject//pubs:AdministrativeMetadata/pubs:DateList/pubs:DatePublished,
ancestor::ddms:resource/ddms:dates/@ddms:posted)"/>
  <sch:param name="flag" value="'error'"/>
</sch:pattern>
```

2.21 - .//Rules/PUBS_ID_00047.sch

Rule Description

[PUBS-ID-00047][Warning] DateValidTil must be later than or equivalent to DateApproved, DatePublished, DatePosted, and DateInfoCutoff. Human Readable: DateValidTil has to be greater than or equal to DateApproved, DatePublished, DatePosted, and DateInfoCutoff.

Code Description

For each ddms:dates element which specifies attribute ddms:validTil, this rule ensures that the date specified in attribute ddms:validTil is later than or equivalent to the date specified in attribute ddms:posted, in attribute ddms:approvedOn, in the ddms:temporalCoverage element with ddms:name [DatePublished], and in the ddms:temporalCoverage element with ddms:name [infoCutOff].

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00047">
  <sch:rule context="tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:dates[@ddms:validTil]">
    <sch:let name="secondaryDateList"
      value="( @approvedOn, ancestor::tdf:*/descendant::pubs:AdministrativeMetadata/pubs:DateList/pubs:DatePublished, @ddms:posted, ancestor::ddms:resource/
ddms:temporalCoverage[ddms:name='infoCutOff']/ddms:approximableEnd/ddms:searchableDate/ddms:end )"/>
    <sch:assert test="every $secondaryDate in $secondaryDateList satisfies dtf:compareDateTimeRanges(string(@ddms:validTil), '>=', string($secondaryDate))"
      flag="warning">[PUBS-ID-00047][Warning] DateValidTil must be later than or equivalent to DateApproved, DatePublished, DatePosted, and DateInfoCutoff. Human
Readable: DateValidTil has to be greater than or equal to DateApproved, DatePublished, DatePosted, and DateInfoCutoff.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.22 - ../Rules/PUBS_ID_00048.sch

Rule Description

[PUBS-ID-00048][Warning] DateRevised must be later than or equivalent to DateApproved, DatePublished, DatePosted, and DateInfoCutoff. Human Readable: DateRevised has to be greater than or equal to DateApproved, DatePublished, DatePosted, and DateInfoCutoff.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the p \$context to each date contained within the p \$secondaryDateList (using the comparison operator contained in p \$operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be found in the abstract pattern definition file located in the Lib directory.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="CompareDateTimes" id="PUBS-ID-00048">
  <sch:param name="ruleText"
    value="' [PUBS-ID-00048][Warning] DateRevised must be later than or equivalent to DateApproved, DatePublished, DatePosted, and DateInfoCutoff. '"/>
  <sch:param name="codeDesc"
    value="' This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $context to each date contained within the
param $secondaryDateList (using the comparison operator contained in param $operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be
found in the abstract pattern definition file located in the Lib directory. '"/>
  <sch:param name="context"
    value="pubs:AdministrativeMetadata/pubs:DateList/pubs:DateRevised"/>
  <sch:param name="operator" value="'>='"/>
  <sch:param name="secondaryDateList"
    value="(ancestor::ddms:resource/ddms:dates/@ddms:approvedOn, ancestor::tdf:*/pubs:AdministrativeMetadata/pubs:DateList/pubs:DatePublished,
ancestor::ddms:resource/ddms:dates/@ddms:posted, ancestor::ddms:resource/ddms:temporalCoverage[ddms:name='infoCutOff']/ddms:approximableEnd/ddms:searchableDate/ddms:end)"/>
  <sch:param name="flag" value="'warning'"/>
</sch:pattern>
```

2.23 - ../Rules/PUBS_ID_00049.sch

Rule Description

[PUBS-ID-00049][Error] DateRevised must be earlier than or equivalent to DateValidTil. Human Readable: DateRevised has to be less than or equal to DateValidTil.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the p \$context to each date contained within the p \$secondaryDateList (using the comparison operator contained in p \$operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be found in the abstract pattern definition file located in the Lib directory.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="CompareDateTimes" id="PUBS-ID-00049">
  <sch:param name="ruleText"
    value="' [PUBS-ID-00049][Error] DateRevised must be earlier than or equivalent to DateValidTil. '"/>
  <sch:param name="codeDesc"
    value="' This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $context to each date contained within the param $secondaryDateList (using the comparison operator contained in param $operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be found in the abstract pattern definition file located in the Lib directory. '"/>
  <sch:param name="context"
    value="pubs:AdministrativeMetadata/pubs:DateList/pubs:DateRevised"/>
  <sch:param name="operator" value="'<='"/>
  <sch:param name="secondaryDateList"
    value="(ancestor::tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:dates/@ddms:validTil)"/>
  <sch:param name="flag" value="'error'"/>
</sch:pattern>
```

2.24 - ../Rules/PUBS_ID_00050.sch

Rule Description

[PUBS-ID-00050][Warning] DateReviewed must be earlier than or equivalent to DateApproved, DatePublished, and DatePosted. Human Readable: DateReviewed has to be less than or equal to DateApproved, DatePublished, and DatePosted.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the p \$context to each date contained within the p \$secondaryDateList (using the comparison operator contained in p \$operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be found in the abstract pattern definition file located in the Lib directory.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="CompareDateTimes" id="PUBS-ID-00050">
  <sch:param name="ruleText"
    value="' [PUBS-ID-00050][Warning] DateReviewed must be earlier than or equivalent to DateApproved, DatePublished, and DatePosted. '"/>
  <sch:param name="codeDesc"
    value="' This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $context to each date contained within the
param $secondaryDateList (using the comparison operator contained in param $operator) and makes sure that each comparison returns true. Implementation details for the abstract pattern can be
found in the abstract pattern definition file located in the Lib directory. '"/>
  <sch:param name="context"
    value="pubs:AdministrativeMetadata/pubs:DateList/pubs:DateReviewed"/>
  <sch:param name="operator" value="'<='"/>
  <sch:param name="secondaryDateList"
    value="(ancestor::tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:dates/@ddms:approvedOn,
ancestor::tdf:*/descendant::pubs:AdministrativeMetadata/pubs:DateList/pubs:DatePublished, ancestor::tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/
tdf:StructuredStatement/ddms:resource/ddms:dates/@ddms:posted)"/>
  <sch:param name="flag" value="'warning'"/>
</sch:pattern>
```


2.25 - .//Rules/PUBS_ID_00053.sch

Rule Description

[PUBS-ID-00053][Error] For any element PostalAddress, at least one of its child elements AddressLine, City, State, Province, PostalCode, or CountryCode must have element content. Human Readable: Element PostalAddress must have a value for at least one of its child elements.

Code Description

For each child element of pubs:PostalAddress listed, this rule normalizes the space of its value and make sures that the length of the resulting string is greater than zero, indicating non-whitespace content.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00053">
    <sch:rule context="pubs:PostalAddress">
        <sch:assert test="some $child in (pubs:AddressLine | pubs:City | pubs:State | pubs:Province | pubs:PostalCode | pubs:CountryCode) satisfies normalize-
space(string($child))"
                                flag="error">[PUBS-ID-00053][Error] For any element PostalAddress, at least one of its child elements AddressLine, City, State, Province, PostalCode, or
CountryCode must have element content. Human Readable: Element PostalAddress must have a value for at least one of its child elements.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.26 - `./Rules/PUBS_ID_00054.sch`

Rule Description

[PUBS-ID-00054][Warning] For elements `ApproximableDateTime`, `ddms:approximableDate`, `ApproximableDateTimeStart`, `ApproximableDateTimeEnd`, `ddms:start`, `ddms:end`, `ddms:approvedOn`, `ddms:postedOn`, `DatePublished`, `ddms:receivedOn`, `DateTimeReferenced`, and `DateInformation`, if the time designator (T) is specified, it is recommended that time zone be specified. Human Readable: It is recommended that time zone be specified if time designator (T) is specified for the following elements: `ApproximableDateTime`, `ddms:approximableDate`, `ApproximableDateTimeStart`, `ApproximableDateTimeEnd`, `ddms:start`, `ddms:end`, `ddms:approvedOn`, `ddms:postedOn`, `DatePublished`, `ddms:receivedOn`, `DateTimeReferenced`, and `DateInformation`.

Code Description

This pattern uses an abstract rule to consolidate logic. If the value of the context contains the time zone designator (T), then it makes sure that the value of the context matches the regular expression for a date with a time zone specified. The abstract rule is extended once for each required element listed in rule `PUBS-ID-00054`.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
    This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00054"><!-- Abstract rule for elements, which asserts that if the time designator (T) is specified, then the timezone is specified -->
<sch:rule abstract="true" id="abs.rule00054">
    <sch:assert test="if (contains(string($valueTest),'T')) then matches(string($valueTest),$endsWithTimeZoneRegEx) else true()"
        flag="warning">[PUBS-ID-00054][Warning] For element
        <sch:name/>, if the time designator (T) is specified, it is recommended that time zone be specified. Human Readable: It is recommended that time zone be specified if time designator (T)
is specified for the following elements: ApproximableDateTime, ddms:approximableDate, ApproximableDateTimeStart, ApproximableDateTimeEnd, ddms:start, ddms:end, ddms:approvedOn, ddms:postedOn,
DatePublished, ddms:receivedOn, DateTimeReferenced, and DateInformation.
    </sch:assert>
    </sch:rule>
    <!-- Abstract rule for attributes, which asserts that if the time designator (T) is specified, then the timezone is specified -->
<sch:rule abstract="true" id="abs.rule00054attrs">
    <sch:let name="fails"
        value="for $date in $dateValues return if(if (contains(string($date),'T')) then matches(string($date),$endsWithTimeZoneRegEx) else true()) then null else
name($date)"/>
    <sch:assert test="count($fails)=0" flag="warning">[PUBS-ID-00054][Warning] For attribute(s):
    <sch:value-of select="for $each in $fails return concat(' ', string($each))"/>if the time designator (T) is specified, it is recommended that the time zone be specified.
    </sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule on required elements -->
<sch:rule context="pubs:ApproximableDateTime">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="ddms:approximableDate">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="pubs:ApproximableDateTimeStart">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="pubs:ApproximableDateTimeEnd">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="pubs:DatePublished">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="pubs:DateTimeReferenced">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="pubs:DateString">
    <sch:let name="valueTest" value="."/>
    <sch:extends rule="abs.rule00054"/>
</sch:rule>
<sch:rule context="pubs:EarliestStartDate">
```

```
        <sch:let name="valueTest" value="."/>
        <sch:extends rule="abs.rule00054"/>
    </sch:rule>
    <sch:rule context="pubs:LatestEndDate">
        <sch:let name="valueTest" value="."/>
        <sch:extends rule="abs.rule00054"/>
    </sch:rule>
</sch:pattern>
```

2.27 - ../Rules/PUBS_ID_00055.sch

Rule Description

[PUBS-ID-00055][Warning] For attributes dateTime, normalizedDateTime, and normalizedTime, it is recommended that time zone be specified. Human Readable: It is recommended that a time zone be specified in the attributes dateTime, normalizedDateTime, and normalizedTime.

Code Description

For attributes dateTime, normalizedDateTime, and normalizedTime, this rule ensures that if the attribute is specified, then it matches the regular expression for a date with a time zone specified.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00055">
    <sch:rule context="pubs:*[@dateTime | @normalizedDateTime | @normalizedTime]">
        <sch:assert test="(if(@dateTime) then matches(string(@dateTime),$endsWithTimeZoneRegEx) else true()) and (if(@normalizedDateTime) then
matches(string(@normalizedDateTime),$endsWithTimeZoneRegEx) else true()) and (if(@normalizedTime) then matches(string(@normalizedTime),$endsWithTimeZoneRegEx) else true())"
            flag="warning">[PUBS-ID-00055][Warning] For attributes dateTime, normalizedDateTime, and normalizedTime, it is recommended that time zone be specified.
Human Readable: It is recommended that a time zone be specified in the attributes dateTime, normalizedDateTime, and normalizedTime.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.28 - ../Rules/PUBS_ID_00062.sch

Rule Description

[PUBS-ID-00062][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of UNECE20 the value of attribute unitOfMeasure must be in CVEnumUNCE20UnitsOfMeasure.xml. Human Readable: If element QuantityReference attribute unitOfMeasureVocabulary value is UNECE20, then the value of attribute unitOfMeasure must be in CVEnumUNCE20UnitsOfMeasure.xml.

Code Description

This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter \$searchTerm is contained in the parameter \$list. The parameter \$searchTerm is relative in scope to the parameter \$context. The value for the parameter \$list is a variable defined in the main document (PUBS_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="ValidateValueExistenceInList" id="PUBS-ID-00062">
    <sch:param name="ruleText"
        value="" [PUBS-ID-00062][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of UNECE20 the value of attribute
unitOfMeasure must be in CVEnumUNCE20UnitsOfMeasure.xml. '"/>
    <sch:param name="codeDesc"
        value="" This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter $searchTerm is contained in the parameter $list. The
parameter $searchTerm is relative in scope to the parameter $context. The value for the parameter $list is a variable defined in the main document (PUBS_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="pubs:QuantityReference[@unitOfMeasureVocabulary='UNECE20']"/>
    <sch:param name="searchTerm" value="@unitOfMeasure"/>
    <sch:param name="list" value="$unce20UnitsOfMeasureList"/>
    <sch:param name="errMsg"
        value="" [PUBS-ID-00062][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of UNECE20 the value of attribute
unitOfMeasure must be in CVEnumUNCE20UnitsOfMeasure.xml. '"/>
</sch:pattern>
```

2.29 - ../Rules/PUBS_ID_00063.sch

Rule Description

[PUBS-ID-00063][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of ISO4217-3 the value of attribute unitOfMeasure must be in CVEnumISO4217Trigraph.xml. Human Readable: If element QuantityReference attribute unitOfMeasureVocabulary value is ISO4217-3, then the value of attribute unitOfMeasure must be in CVEnumISO4217Trigraph.xml.

Code Description

This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter \$searchTerm is contained in the parameter \$list. The parameter \$searchTerm is relative in scope to the parameter \$context. The value for the parameter \$list is a variable defined in the main document (PUBS_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="ValidateValueExistenceInList" id="PUBS-ID-00063">
    <sch:param name="ruleText"
        value="" [PUBS-ID-00063][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of ISO4217-3 the value of attribute
unitOfMeasure must be in CVEnumISO4217Trigraph.xml. '"/>
    <sch:param name="codeDesc"
        value="" This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter $searchTerm is contained in the parameter $list. The
parameter $searchTerm is relative in scope to the parameter $context. The value for the parameter $list is a variable defined in the main document (PUBS_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="pubs:QuantityReference[@unitOfMeasureVocabulary='ISO4217-3']"/>
    <sch:param name="searchTerm" value="@unitOfMeasure"/>
    <sch:param name="list" value="$iso4217TrigraphList"/>
    <sch:param name="errMsg"
        value="" [PUBS-ID-00063][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of ISO4217-3 the value of attribute
unitOfMeasure must be in CVEnumISO4217Trigraph.xml. '"/>
</sch:pattern>
```

2.30 - ../Rules/PUBS_ID_00064.sch

Rule Description

[PUBS-ID-00064][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of ISO4217-NR the value of attribute unitOfMeasure must be in CVENumISO4217Numeric.xml. Human Readable: If element QuantityReference attribute unitOfMeasureVocabulary value is ISO4217-NR, then the value of attribute unitOfMeasure must be in CVENumISO4217Numeric.xml.

Code Description

This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter \$searchTerm is contained in the parameter \$list. The parameter \$searchTerm is relative in scope to the parameter \$context. The value for the parameter \$list is a variable defined in the main document (PUBS_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="ValidateValueExistenceInList" id="PUBS-ID-00064">
    <sch:param name="ruleText"
        value="" [PUBS-ID-00064][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of ISO4217-NR the value of attribute
unitOfMeasure must be in CVENumISO4217Numeric.xml. '"/>
    <sch:param name="codeDesc"
        value="" This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter $searchTerm is contained in the parameter $list. The
parameter $searchTerm is relative in scope to the parameter $context. The value for the parameter $list is a variable defined in the main document (PUBS_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="pubs:QuantityReference[@unitOfMeasureVocabulary='ISO4217-NR']"/>
    <sch:param name="searchTerm" value="@unitOfMeasure"/>
    <sch:param name="list" value="$iso4217NumericList"/>
    <sch:param name="errMsg"
        value="" [PUBS-ID-00064][Error] If the element QuantityReference has the attribute unitOfMeasureVocabulary with a value of ISO4217-NR the value of attribute
unitOfMeasure must be in CVENumISO4217Numeric.xml. '"/>
    </sch:pattern>
```


2.31 - ../Rules/PUBS_ID_00065.sch

Rule Description

[PUBS-ID-00065][Error] For elements: AuthorInfo, CoauthorInfo, ContributingAuthorInfo, POCinfo at least one of the following child elements must have non-whitespace content: Surname, UserID, JobTitle, Affiliation, OfficeName, PostalAddress, PhoneNumber, FaxNumber, EmailAddress, WebPageAddress, FormattedSignatureBlock. Human Readable: Elements AuthorInfo, CoauthorInfo, ContributingAuthorInfo, POCinfo must have a value for at least one of the following child elements: Surname, UserID, JobTitle, Affiliation, OfficeName, PostalAddress, PhoneNumber, FaxNumber, EmailAddress, WebPageAddress, FormattedSignatureBlock.

Code Description

This pattern uses an abstract rule to consolidate logic. It normalizes the space of the value of the specified child elements and makes sure that the length of the resulting string is greater than zero, which indicates non-whitespace content. Element PostalAddress cannot contain text content, so the rule counts the number of its child elements that contain non-white space and makes sure that the count is great than 0. The abstract rule is extended once for each required element in rule PUBS_ID_00065.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00065"><!-- Abstract rule, which asserts that at least one of the listed child elements has non-whitespace content -->
<sch:rule abstract="true" id="abs.rule00065">
    <sch:assert test="pubs:Surname[normalize-space(string(text()))] or pubs:UserID[normalize-space(string(text()))] or pubs:JobTitle[normalize-space(string(text()))] or
pubs:Affiliation[normalize-space(string(text()))] or pubs:OfficeName[normalize-space(string(text()))] or pubs:PhoneNumber[normalize-space(string(text()))] or pubs:FaxNumber[normalize-
space(string(text()))] or pubs:EmailAddress[normalize-space(string(text()))] or pubs:WebPageAddress[normalize-space(string(text()))] or pubs:FormattedSignatureBlock[normalize-
space(string(text()))] or (some $token in pubs:PostalAddress/*/text() satisfies normalize-space(string($token)))"
        flag="error">[PUBS-ID-00065][Error] For element
    <sch:name/>at least one of the following child elements must have non-whitespace content: Surname, UserID, JobTitle, Affiliation, OfficeName, PostalAddress, PhoneNumber, FaxNumber,
EmailAddress, WebPageAddress, FormattedSignatureBlock. Human Readable: Elements AuthorInfo, CoauthorInfo, ContributingAuthorInfo, POCinfo must have a value for at least one of the following
child elements: Surname, UserID, JobTitle, Affiliation, OfficeName, PostalAddress, PhoneNumber, FaxNumber, EmailAddress, WebPageAddress, FormattedSignatureBlock.
    </sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="pubs:AuthorInfo">
    <sch:extends rule="abs.rule00065"/>
</sch:rule>
<sch:rule context="pubs:CoauthorInfo">
    <sch:extends rule="abs.rule00065"/>
</sch:rule>
<sch:rule context="pubs:ContributingAuthorInfo">
    <sch:extends rule="abs.rule00065"/>
</sch:rule>
<sch:rule context="pubs:POCinfo">
    <sch:extends rule="abs.rule00065"/>
</sch:rule>
</sch:pattern>
```

2.32 - ../Rules/PUBS_ID_00070.sch

Rule Description

[PUBS-ID-00070][Error] Element ddms:type must not have the attribute ddms:qualifier with a value of RevisionRecallDesignations. The required practice for conveying Revision Recall is to use a RevisionRecall handling assertion introduced in the 2014 DEC Revision Recall specification.

This rule was originally created for PUBS v10 to prevent the PUBS v9 practice of using the pubs:OtherProperty element to express RevRecall. PUBS v10 directly supported RevRecall, so this practice was prohibited. PUBS 2014-DEC no longer supports RevRecall directly; it uses RevRecall handling assertion.

Code Description

If any element ddms:type has attribute ddms:qualifier specified with a value of [RevisionRecallDesignations], then this rule asserts false (generate an error).

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00070">
  <sch:rule context="tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='RevisionRecallDesignations']">
    <sch:assert test="false()" flag="error">[PUBS-ID-00070][Error] Element ddms:type must not have the attribute ddms:qualifier with a value of RevisionRecallDesignations.
The required practice for conveying Revision Recall is to use a RevisionRecall handling assertion introduced in the 2014 DEC Revision Recall specification.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.33 - ../Rules/PUBS_ID_00072.sch

Rule Description

[PUBS-ID-00072][Error] If element CountryName has attribute countryCodeVocabulary specified as ISO-3 the attribute countryCode value must be in CVENumPubsCountryISO3166Trigraph.xml. Human Readable: If element CountryName attribute countryCodeVocabulary value is ISO-3, then the value of attribute countryCode must be in CVENumPubsCountryISO3166Trigraph.xml.

Code Description

This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter \$searchTerm is contained in the parameter \$list. The parameter \$searchTerm is relative in scope to the parameter \$context. The value for the parameter \$list is a variable defined in the main document (PUBS_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern is-a="ValidateValueExistenceInList" id="PUBS-ID-00072">
    <sch:param name="ruleText"
        value="" [PUBS-ID-00072][Error] If element CountryName has attribute countryCodeVocabulary specified as ISO-3 the attribute countryCode value must be in
CVENumPubsCountryISO3166Trigraph.xml ""/>
    <sch:param name="codeDesc"
        value="" This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter $searchTerm is contained in the parameter $list. The
parameter $searchTerm is relative in scope to the parameter $context. The value for the parameter $list is a variable defined in the main document (PUBS_XML.sch), which reads values from a
specific CVE file. ""/>
    <sch:param name="context" value="pubs:CountryName[@countryCodeVocabulary='ISO-3']"/>
    <sch:param name="searchTerm" value="@countryCode"/>
    <sch:param name="list" value="$countryIso3166TrigraphList"/>
    <sch:param name="errMsg"
        value="" [PUBS-ID-00072][Error] If element CountryName has attribute countryCodeVocabulary specified as ISO-3 the attribute countryCode value must be in
CVENumPubsCountryISO3166Trigraph.xml ""/>
    </sch:pattern>
```

2.34 - .//Rules/PUBS_ID_00079.sch

Rule Description

[PUBS-ID-00079][Error] Element DescriptiveMetadata/Title must not start with any of the following strings ADMINISTRATIVE RECALL, or ADMINISTRATIVE REVISION, or SUBSTANTIVE RECALL, or SUBSTANTIVE REVISION. Human Readable: Using text in the Title is incorporated into the schema using RevisionRecall. Element DescriptiveMetadata/Title cannot start with any of the following strings: ADMINISTRATIVE RECALL, or ADMINISTRATIVE REVISION, or SUBSTANTIVE RECALL, or SUBSTANTIVE REVISION.

Code Description

If element DescriptiveMetadata/Title is specified with a value that starts with ADMINISTRATIVE RECALL, ADMINISTRATIVE REVISION, SUBSTANTIVE RECALL, or SUBSTANTIVE REVISION, then this rule returns false. The Revision Recall Best practice method of using text in the Title is now incorporated into the schema using RevisionRecall therefore use in this DES is rescinded.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00079">
    <sch:rule context="pubs:DescriptiveMetadata/pubs:Title | tdf:*[descendant::tdf:StructuredPayload[pubs:IntelDoc]]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:title">
        <sch:assert test="if(starts-with(upper-case(.),'ADMINISTRATIVE RECALL') or starts-with(upper-case(.),'ADMINISTRATIVE REVISION') or starts-with(upper-case(.),'SUBSTANTIVE RECALL') or starts-with(upper-case(.),'SUBSTANTIVE REVISION') ) then false() else true()"
            flag="error">[PUBS-ID-00079][Error] Element DescriptiveMetadata/Title must not start with any of the following strings ADMINISTRATIVE RECALL, or ADMINISTRATIVE REVISION, or SUBSTANTIVE RECALL, or SUBSTANTIVE REVISION. Human Readable: Element DescriptiveMetadata/Title cannot start with any of the following strings: ADMINISTRATIVE RECALL, or ADMINISTRATIVE REVISION, or SUBSTANTIVE RECALL, or SUBSTANTIVE REVISION.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.35 - ../Rules/PUBS_ID_00080.sch

Rule Description

[PUBS-ID-00080][Error] The attribute `ism:excludeFromRollup` must not be specified for any element in the namespace `urn:us:gov:ic:pubs` except `ReferencedResourceSecurity`. Human Readable: Attribute `ism:excluedFromRollup` is only permitted with element `ReferenceResourceSecurity` in the `urn:us:gov:ic:pubs` namespace.

Code Description

For all elements in the namespace `urn:us:gov:ic:pubs` with attribute `ism:excludeFromRollup` specified, this rule ensures that the name of the element is `ReferencedResourceSecurity`.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00080">
    <sch:rule context="pubs:*[@ism:excludeFromRollup]">
        <sch:assert test="self::pubs:ReferencedResourceSecurity" flag="error">[PUBS-ID-00080][Error] The attribute ism:excludeFromRollup must not be specified for any element
in the namespace urn:us:gov:ic:pubs except ReferencedResourceSecurity. Human Readable: Attribute ism:excluedFromRollup is only permitted with element ReferenceResourceSecurity in the
urn:us:gov:ic:pubs namespace.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.36 - ../Rules/PUBS_ID_00081.sch

Rule Description

[PUBS-ID-00081][Error] If element SourceReference has attribute publisherType with a value of [open-source], then child element SourceType must be specified. Human Readable: If element SourceReference attribute publisherType has a value of [open-source], then there must be a value for child element SourceType.

Code Description

For each element SourceReference with attribute publisherType specified with a value of [open-source], this rule normalizes the space of the value of child element SourceType and makes sure that the length of the resulting string is greater than zero, which indicates non-whitespace content.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00081">
    <sch:rule context="pubs:SourceReference[@publisherType='open-source']">
        <sch:assert test="string-length(normalize-space(string(../pubs:SourceType))) > 0"
            flag="error">[PUBS-ID-00081][Error] If element SourceReference has attribute publisherType with a value of [open-source], then child element SourceType must
be specified. Human Readable: If element SourceReference attribute publisherType has a value of [open-source], then there must be a value for child element SourceType.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.37 - ./Rules/PUBS_ID_00082.sch

Rule Description

[PUBS-ID-00082][Error] The attribute ism:noticeType must not be specified for any element in the namespace urn:us:gov:ic:pubs except IntelDoc, Note, NoteInline, AuthorInfo, CoauthorInfo, ContributingAuthorInfo, or POCinfo. Human Readable: Attribute ism:noticeType can only exist in the namespace urn:us:gov:ic:pubs for the following elements: IntelDoc, Note, NoteInline, AuthorInfo, CoauthorInfo, ContributingAuthorInfo, or POCinfo.

Code Description

For all elements in the namespace urn:us:gov:ic:pubs with attribute ism:noticeType specified, this rule ensures that the name of the element is one of IntelDoc, Note, NoteInline, AuthorInfo, CoauthorInfo, ContributingAuthorInfo, or POCinfo.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00082">
  <sch:rule context="pubs:*[@ism:noticeType][not(self::pubs:Note | self::pubs:AuthorInfo | self::pubs:NoteInline | self::pubs:IntelDoc | self::pubs:CoauthorInfo |
self::pubs:ContributingAuthorInfo | self::pubs:POCinfo)]">
    <sch:assert test="false()" flag="error">[PUBS-ID-00082][Error] The attribute ism:noticeType must not be specified for any element in the namespace urn:us:gov:ic:pubs
except IntelDoc, Note, NoteInline, AuthorInfo, CoauthorInfo, ContributingAuthorInfo, or POCinfo. Human Readable: Attribute ism:noticeType can only exist in the namespace urn:us:gov:ic:pubs for
the following elements: IntelDoc, Note, NoteInline, AuthorInfo, CoauthorInfo, ContributingAuthorInfo, or POCinfo.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.38 - ../Rules/PUBS_ID_00086.sch

Rule Description

[PUBS-ID-00086][Error] If PublicationMetadataList contains more than one PublicationMetadata then all the PublicationMetadata elements must have ISM security marks present.

Code Description

For each PublicationMetadataList element with more than one PublicationMetadata child, we check that the PublicationMetadata child element specifies the attribute ism:classification, which indicates ISM security markings.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00086">
    <sch:rule context="pubs:PublicationMetadataList[count(pubs:PublicationMetadata) > 1]/pubs:PublicationMetadata">
        <sch:assert test="@ism:classification" flag="error">[PUBS-ID-00086][Error] When there are multiple PublicationMetadata each must be security portion marked to
facilitate use of PublicationMetadata at a different level than the document.</sch:assert>
    </sch:rule>
</sch:pattern>
```


2.39 - ../Rules/PUBS_ID_00087.sch

Rule Description

[PUBS-ID-00087][Error] If the DateInformation element exists, at least one of its child elements DateString, ApproximableDateTime, or SearchableDateTime must be present. Human Readable: Element DateInformation must have a value for one of its child elements.

Code Description

For element pubs:DateInformation, this rule ensures that one or more of the child elements DateString, ApproximableDateTime, SearchableDateTime/EarliestStartDate or SearchableDateTime/LatestEndDate is specified with a non-white space value.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00087">
    <sch:rule context="pubs:DateInformation">
        <sch:assert test="normalize-space(string(pubs:DateString)) or normalize-space(string(pubs:ApproximableDateTime)) or normalize-space(string(pubs:SearchableDateTime/
pubs:EarliestStartDate)) or normalize-space(string(pubs:SearchableDateTime/pubs:LatestEndDate))"
            flag="error">[PUBS-ID-00087][Error] If the DateInformation element exists, at least one of its child elements DateString, ApproximableDateTime, or
SearchableDateTime must be present. Human Readable: Element DateInformation must have a value for one of its child elements.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.40 - ../Rules/PUBS_ID_00092.sch

Rule Description

[PUBS-ID-00092][Error] For element DateInformation, the permissible values for the year range are 0001 through 9999 for elements ApproximableDateTime, SearchableDateTime/EarliestStartDate, and SearchableDateTime/LatestEndDate. Human Readable: For element DateInformation, year values for elements ApproximableDateTime, SearchableDateTime/EarliestStartDate, and SearchableDateTime/LatestEndDate have to fall within the range 0001 through 9999.

Code Description

This pattern uses an abstract rule to consolidate logic. It makes sure that the date contained within \$dateValue has a year value within the range \$minYear and \$maxYear, inclusive. The abstract rule is extended once for each element required in rule PUBS-ID-00092.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00092">
    <sch:rule abstract="true" id="abs.rule00092">
        <sch:let name="minYear" value="0001"/>
        <sch:let name="maxYear" value="9999"/>
        <sch:let name="dateValue" value="."/>
        <sch:let name="errMsg"
            value="" [PUBS-ID-00092][Error] For element DateInformation, the permissible values for the year range are 0001 through 9999 for elements ApproximableDateTime,
SearchableDateTime/EarliestStartDate, and SearchableDateTime/LatestEndDate. '"/>
        <sch:extends rule="abs.dateYearRangeRule"/>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="pubs:DateInformation/pubs:ApproximableDateTime">
    <sch:extends rule="abs.rule00092"/>
</sch:rule>
<sch:rule context="pubs:DateInformation/pubs:SearchableDateTime/pubs:EarliestStartDate">
    <sch:extends rule="abs.rule00092"/>
</sch:rule>
<sch:rule context="pubs:DateInformation/pubs:SearchableDateTime/pubs:LatestEndDate">
    <sch:extends rule="abs.rule00092"/>
</sch:rule>
</sch:pattern>
```

2.41 - ../Rules/PUBS_ID_00093.sch

Rule Description

[PUBS-ID-00093][Error] Every attribute on a PUBS element in the document must be specified with a non-whitespace value.

Code Description

For each PUBS element with at least one attribute specified, this rule ensures that all attributes on that element are specified with a non-whitespace value;

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00093">
    <sch:rule context="pubs:*[*]">
        <sch:assert test="every $attribute in @* satisfies normalize-space(string($attribute))"
                    flag="error">[PUBS-ID-00093][Error] Every attribute on a PUBS element in the document must be specified with a non-whitespace value.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.42 - ../Rules/PUBS_ID_00096.sch

Rule Description

[PUBS-ID-00096][Error] For elements pubs:Facility and pubs:Person, if attribute xlink:type is specified then it must have a value of [simple] or [resource]. Human Readable: For elements pubs:Facility and pubs:Person, attribute xlink:type value must be either [simple] or [resource].

Code Description

Makes sure that for each element pubs:Facility and pubs:Person, if attribute xlink:type is specified then it has a value of [simple] or [resource].

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00096">
    <sch:rule context="pubs:Facility[@xlink:type] | pubs:Person[@xlink:type]">
        <sch:assert test="@xlink:type='simple' or @xlink:type='resource'" flag="error">[PUBS-ID-00096][Error] For elements pubs:Facility and pubs:Person, if attribute
xlink:type is specified then it must have a value of [simple] or [resource]. Human Readable: For elements pubs:Facility and pubs:Person, attribute xlink:type value must be either [simple] or
[resource].</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.43 - ../Rules/PUBS_ID_00097.sch

Rule Description

[PUBS-ID-00097][Error] For element pubs:Link, if attribute xlink:type is defined then it must have a value of [simple] or [extended]. Human Readable: For element pubs:Link, attribute xlink:type value must be either [simple] or [extended].

Code Description

Makes sure that for each element pubs:Link, if attribute xlink:type is specified then it has a value of [simple] or [extended].

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00097">
    <sch:rule context="pubs:Link[@xlink:type]">
        <sch:assert test="@xlink:type='simple' or @xlink:type='extended'" flag="error">[PUBS-ID-00097][Error] For element pubs:Link, if attribute xlink:type is defined then it
must have a value of [simple] or [extended]. Human Readable: For element pubs:Link, attribute xlink:type value must be either [simple] or [extended].</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.44 - ../Rules/PUBS_ID_00098.sch

Rule Description

[PUBS-ID-00098][Error] For any element, if attribute xlink:type is specified with a value of [extended], then that element can only define the following attributes in the xlink namespace: xlink:type, xlink:role, and xlink:title. Human Readable: If any element has attribute xlink:type with a value of [extended], then that element can only have values for the following in the xlink namespace: xlink:type, xlink:role, and xlink:title.

Code Description

Makes sure that for each element that has attribute xlink:type specified with a value of [extended] only defines the following attributes in the xlink namespace: xlink:type, xlink:role, and xlink:title.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00098">
    <sch:rule context="pubs:*[@xlink:type='extended']">
        <sch:let name="xlinkAttrLocalNames"
            value="for $attr in @xlink:* return $attr/local-name()"/>
        <sch:let name="allowedXlinkLocalNamesList" value="('type', 'role', 'title')"/>
        <sch:assert test="every $attrLocalName in $xlinkAttrLocalNames satisfies index-of($allowedXlinkLocalNamesList, $attrLocalName)"
            flag="error">[PUBS-ID-00098][Error] For any element, if attribute xlink:type is specified with a value of [extended], then that element can only define the
following attributes in the xlink namespace: xlink:type, xlink:role, and xlink:title. Human Readable: If any element has attribute xlink:type with a value of [extended], then that element can
only have values for the following in the xlink namespace: xlink:type, xlink:role, and xlink:title.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.45 - ../Rules/PUBS_ID_00099.sch

Rule Description

[PUBS-ID-00099][Error] For any element, if attribute xlink:type is specified with a value of [resource], then that element can only define the following attributes in the xlink namespace: xlink:type, xlink:role, xlink:title, and xlink:label. Human Readable: If any element has attribute xlink:type with a value of [resource], then that element can only have values for the following in the xlink namespace: xlink:type, xlink:role, xlink:title, and xlink:label.

Code Description

Makes sure that for each element that has attribute xlink:type specified with a value of [extended] only defines the following attributes in the xlink namespace: xlink:type, xlink:role, and xlink:title.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00099">
    <sch:rule context="pubs:*[@xlink:type='resource']">
        <sch:let name="xlinkAttrLocalNames"
            value="for $attr in @xlink:* return $attr/local-name()"/>
        <sch:let name="allowedXlinkLocalNamesList"
            value="('type', 'role', 'title', 'label')"/>
        <sch:assert test="every $attrLocalName in $xlinkAttrLocalNames satisfies index-of($allowedXlinkLocalNamesList, $attrLocalName)"
            flag="error">[PUBS-ID-00099][Error] For any element, if attribute xlink:type is specified with a value of [resource], then that element can only define the
following attributes in the xlink namespace: xlink:type, xlink:role, xlink:title, and xlink:label. Human Readable: If any element has attribute xlink:type with a value of [resource], then that
element can only have values for the following in the xlink namespace: xlink:type, xlink:role, xlink:title, and xlink:label.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.46 - ../Rules/PUBS_ID_00100.sch

Rule Description

[PUBS-ID-00100][Error] For any element, if attribute xlink:type is specified with a value of [simple], then that element can only define the following attributes in the xlink namespace: xlink:type, xlink:href, xlink:role, xlink:arcrole, xlink:title, xlink:show, and xlink:actuate. Human Readable: If any element has attribute xlink:type with a value of [simple], then that element can only have values for the following in the xlink namespace: xlink:type, xlink:href, xlink:role, xlink:arcrole, xlink:title, xlink:show, and xlink:actuate.

Code Description

Makes sure that for each element that has attribute xlink:type specified with a value of [extended] only defines the following attributes in the xlink namespace: xlink:type, xlink:role, and xlink:title.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00100">
    <sch:rule context="pubs:*[@xlink:type='simple']">
        <sch:let name="xlinkAttrLocalNames"
            value="for $attr in @xlink:* return $attr/local-name()"/>
        <sch:let name="allowedXlinkLocalNamesList"
            value="('type', 'href', 'role', 'arcrole', 'title', 'show', 'actuate')"/>
        <sch:assert test="every $attrLocalName in $xlinkAttrLocalNames satisfies index-of($allowedXlinkLocalNamesList, $attrLocalName)"
            flag="error">[PUBS-ID-00100][Error] For any element, if attribute xlink:type is specified with a value of [simple], then that element can only define the
following attributes in the xlink namespace: xlink:type, xlink:href, xlink:role, xlink:arcrole, xlink:title, xlink:show, and xlink:actuate. Human Readable: If any element has attribute
xlink:type with a value of [simple], then that element can only have values for the following in the xlink namespace: xlink:type, xlink:href, xlink:role, xlink:arcrole, xlink:title,
xlink:show, and xlink:actuate.</sch:assert>
    </sch:rule>
</sch:pattern>
```


2.47 - ../Rules/PUBS_ID_00101.sch

Rule Description

[PUBS-ID-00101][Error] For any element, if any attribute is specified with the xlink namespace 'http://www.w3.org/1999/xlink', then attributes xlink:type and/or xlink:href must be specified. Human Readable: If any element has an attribute specified with the xlink namespace 'http://www.w3.org/1999/xlink', then attributes xlink:type and/or xlink:href must be specified.

Code Description

Makes sure that for each element that has any attribute in the xlink namespace has either xlink:type or xlink:href specified.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00101">
    <sch:rule context="pubs:*[@xlink:*)">
        <sch:assert test="normalize-space(string(@xlink:type)) or normalize-space(string(@xlink:href))"
            flag="error">[PUBS-ID-00101][Error] For any element, if any attribute is specified with the xlink namespace 'http://www.w3.org/1999/xlink', then attributes
xlink:type and/or xlink:href must be specified. Human Readable: If any element has an attribute specified with the xlink namespace 'http://www.w3.org/1999/xlink', then attributes xlink:type
and/or xlink:href must be specified.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.48 - ../Rules/PUBS_ID_00103.sch

Rule Description

[PUBS-ID-00103][Error] If an element has @noteType then it cannot have @ism:noticeType or @ism:unregisteredNoticeType. Human Readable: An element with attribute noteType cannot have the ISM attributes noticeType or unregisteredNoticeType.

Code Description

If an element has a noteType attribute, then it may not have the ISM attributes noticeType or unregisteredNoticeType.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00103">
    <sch:rule context="pubs:*[@noteType]">
        <sch:assert test="not(@ism:unregisteredNoticeType|@ism:noticeType)" flag="error">[PUBS-ID-00103][Error] If an element has @noteType then it cannot have @ism:noticeType
or @ism:unregisteredNoticeType. Human Readable: An element with attribute noteType cannot have the ISM attributes noticeType or unregisteredNoticeType.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.49 - ../Rules/PUBS_ID_00104.sch

Rule Description

[PUBS-ID-00104][Error] When there is a mixture of Foreign Government Information (FGI) and US data present on a Para element with ism:ownerProducer containing [USA], then there must exist a source citation reference. Human Readable: Element Para with ism:ownerProducer containing [USA] and a mix of Foreign Government Information (FGI) and US data, must have a source citation reference.

Code Description

Given a pubs:Para element with @ism:ownerProducer containing USA and either attribute @ism:FGIsourceOpen or @ism:FGIsourceProtected, there must exist a source citation reference (pubs:SourceReferenceCitationRef).

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00104">
    <sch:rule context="pubs:Para[contains(concat(' ',string(@ism:ownerProducer),' '), ' USA ')] and (@ism:FGIsourceOpen or @ism:FGIsourceProtected)]">
        <sch:assert test="../pubs:SourceReferenceCitationRef" flag="error">[PUBS-ID-00104][Error] When there is a mixture of Foreign Government Information (FGI) and US data
present on a Para element with ism:ownerProducer containing [USA], then there must exist a source citation. reference Human Readable: Element Para with ism:ownerProducer containing [USA] and a
mix of Foreign Government Information (FGI) and US data, must have a source citation reference.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.50 - ../Rules/PUBS_ID_00110.sch

Rule Description

[PUBS-ID-00110][Error] In PUBS, the attribute ddms:compliesWith on element ddms:resource must always have complies with specified with a value of [MIN_DISCOVERABLE]. Human Readable: Element ddms:resource attribute ddms:compliesWith requires a value of MIN_DISCOVERABLE.

Code Description

Make sure that ddms:resource has attribute ddms:compliesWith specified with a value of MIN_DISCOVERABLE.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00110">
    <sch:rule context="tdf:*[descendant::tdf:StructuredPayload/pubs:IntelDoc]/tdf:Assertion/tdf:StructuredStatement/ddms:resource">
        <sch:assert test="@ddms:compliesWith='MIN_DISCOVERABLE'" flag="error">[PUBS-ID-00110][Error] In PUBS, the attribute ddms:compliesWith on element ddms:resource must
always have complies with specified with a value of [MIN_DISCOVERABLE]. Human Readable: Element ddms:resource attribute ddms:compliesWith requires a value of MIN_DISCOVERABLE.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.51 - ../Rules/PUBS_ID_00111.sch

Rule Description

[PUBS-ID-00111][Error] At least one of either ddms:resource/ddms:dates/@ddms:posted or pubs:DatePublished is required to be specified. Human Readable: A PUBS document must have either a date posted or date published specified for itself.

Code Description

If a TDO has a PUBS assertion, determined by checking if it has a tdf:StructuredStatement element with a child of pubs:IntelDocMetadata, this rule verifies that at least one of tdf:Assertion//ddms:resource/ddms:dates/@ddms:posted or tdf:StructuredPayload//pubs:DatePublished is specified.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00111">
    <sch:rule context="tdf:TrustedDataObject[descendant::tdf:StructuredStatement/pubs:IntelDocMetadata]">
        <sch:assert test="tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:dates[@ddms:posted] or tdf:StructuredPayload/descendant::pubs:DatePublished"
            flag="error">[PUBS-ID-00111][Error] At least one of either ddms:resource/ddms:dates/@ddms:posted or pubs:DatePublished is required to be specified. Human
Readable: A PUBS document must have either a date posted or date published specified for itself.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.52 - ../Rules/PUBS_ID_00113.sch

Rule Description

[PUBS-ID-00113][Error] If a TrustedDataObject element contains an assertion with a PUBS structured statement, it must also contain an assertion with an IRM structured statement. Human Readable: If a Trusted Data Object contains a PUBS assertion, it must also contain an IRM assertion.

Code Description

If a TDO has a PUBS assertion, determined by checking if it has a tdf:StructuredStatement element with a child of pubs:IntelDocMetadata, verify that it also contains an IRM assertion, verified by checking if it has a tdf:StructuredStatement element with a child of irm:ICResourceMetadataPackage.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00113">
    <sch:rule context="tdf:TrustedDataObject[descendant::tdf:StructuredStatement/pubs:IntelDocMetadata]">
        <sch:assert test="descendant::tdf:StructuredStatement/irm:ICResourceMetadataPackage"
            flag="error">[PUBS-ID-00113][Error] If tdf:TrustedDataObject contains an assertion with a PUBS structured statement, it must also contain an assertion with
an IRM structured statement. Human Readable: If a Trusted Data Object contains a PUBS assertion, it must also contain an IRM assertion.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.53 - ../Rules/PUBS_ID_00114.sch

Rule Description

[PUBS-ID-00114][Error] The @ism:DESVersion is less than the minimum version allowed: 13. Human Readable: The ISM version imported by PUBS must be greater than or equal to 13.

Code Description

For all elements that contain @ism:DESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 13.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00114">
  <sch:rule context="*[@ism:DESVersion]">
    <sch:let name="version"
      value="number(if (contains(@ism:DESVersion,'-')) then substring-before(@ism:DESVersion,'-') else @ism:DESVersion)"/>
    <sch:assert test="$version >= 13" flag="error">[PUBS-ID-00114][Error] The @ism:DESVersion is less than the minimum version allowed: 13. Human Readable: The ISM
version imported by PUBS must be greater than or equal to 13.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.54 - ../Rules/PUBS_ID_00115.sch

Rule Description

[PUBS-ID-00115][Error] The root node of a PUBS document must be pubs:IntelDoc in a tdf:StructuredPayload element of a TDF instance. Human Readable: PUBS must be in a TDF and the starting node in the payload must be pubs:IntelDoc.

Code Description

Ensure that the root node of the document is in the tdf namespace and that the beginning of a structured payload consists for a pubs:IntelDoc element.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00115">
  <sch:rule context="/">
    <sch:assert test="tdf:*//tdf:StructuredPayload/pubs:IntelDoc" flag="error">[PUBS-ID-00115][Error] The root node of a PUBS document must be pubs:IntelDoc in a
tdf:StructuredPayload element of a TDF instance. Human Readable: PUBS must be in a TDF and the starting node in the payload must be pubs:IntelDoc.</sch:assert>
  </sch:rule>
</sch:pattern>
```


2.55 - ../Rules/PUBS_ID_00116.sch

Rule Description

[PUBS-ID-00116][Error] If a tdf:TrustedDataObject has an IntelDoc element, it must contain a pubs:IntelDocMetadata assertion.

Code Description

A PUBS Trusted Data Object IntelDoc must contain a PUBS assertion.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00116">
    <sch:rule context="tdf:TrustedDataObject/tdf:StructuredPayload/pubs:IntelDoc">
        <sch:assert test="ancestor::tdf:TrustedDataObject/tdf:Assertion/tdf:StructuredStatement/pubs:IntelDocMetadata"
                    flag="error">[PUBS-ID-00116][Error] If a tdf:TrustedDataObject has an IntelDoc element, it must contain a pubs:IntelDocMetadata assertion.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.56 - ../Rules/PUBS_ID_00117.sch

Rule Description

[PUBS-ID-00117][Error] The value of each attribute @src:citationIDRef must be found in the list of attribute @src:citationID values.

Code Description

The value of each attribute @src:citationIDRef must be found in the list of attribute @src:citationID values.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00117">
  <sch:rule context="pubs:SourceReferenceCitationRef|pubs:AppendedReferenceCitationRef">
    <sch:let name="citationIDList"
      value="ancestor::tdf:TrustedDataObject//@src:citationID"/>
    <sch:assert test="some $citationID in $citationIDList satisfies compare($citationID,normalize-space(@src:citationIDRef)) = 0"
      flag="error">[PUBS-ID-00117][Error] The value of each attribute @src:citationIDRef (
    <sch:value-of select="@src:citationIDRef"/>) must be found in the list of attribute @src:citationID values. (
    <sch:value-of select="for $citationID in $citationIDList return $citationID"/>)
  </sch:assert>
  </sch:rule>
</sch:pattern>
```

2.57 - ../Rules/PUBS_ID_00118.sch

Rule Description

[PUBS-ID-00118][Error] The @mime:CESVersion is less than the minimum version allowed: 201609. Human Readable: The MIME version imported by PUBS must be greater than or equal to 2016-SEP.

Code Description

For all elements that contain @mime:CESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 201609.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00118">
  <sch:rule context="*[@mime:CESVersion]">
    <sch:let name="version"
      value="number(if (contains(@mime:CESVersion,'-')) then substring-before(@mime:CESVersion,'-') else @mime:CESVersion)"/>
    <sch:assert test="$version >= 201609" flag="error">[PUBS-ID-00118][Error] The @mime:CESVersion is less than the minimum version allowed: 201609. Human Readable: The
MIME version imported by PUBS must be greater than or equal to 2016-SEP.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.58 - ../Rules/PUBS_ID_00119.sch

Rule Description

[PUBS-ID-00119][Error] If @mime:mimeType exists, then @mime:CESVersion must exist as well.

Code Description

Make sure that the MIME CVE version attribute exists if MIME mimeType exists.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00119">
    <sch:rule context="*[pubs:MediaResource//@mime:mimeType]">
        <sch:assert test="//@mime:CESVersion" flag="error">[PUBS-ID-00119][Error] If @mime:mimeType exists, then @mime:CESVersion must exist as well.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.59 - ../Rules/PUBS_ID_00120.sch

Rule Description

[PUBS-ID-00120][Error] The @intdis:CESVersion is less than the minimum version allowed: 201508. Human Readable: The INTDIS version imported by PUBS must be greater than or equal to 2015-AUG.

Code Description

For all elements that contain @intdis:CESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 201508.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00120">
  <sch:rule context="pubs:IntelDocMetadata[@intdis:CESVersion]">
    <sch:let name="version"
      value="number(if (contains(@intdis:CESVersion,'-')) then substring-before(@intdis:CESVersion,'-') else @intdis:CESVersion)"/>
    <sch:assert test="$version >= 201508" flag="error">[PUBS-ID-00120][Error] The @intdis:CESVersion is less than the minimum version allowed: 201508. Human Readable:
The INTDIS version imported by PUBS must be greater than or equal to 2015-AUG.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.60 - ../Rules/PUBS_ID_00121.sch

Rule Description

[PUBS-ID-00121][Error] If pubs:IntelDiscipline exists, then @intdis:CESVersion must exist as well.

Code Description

Make sure that the INTDIS CVE version attribute exists if PUBS IntelDiscipline exists.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00121">
    <sch:rule context="pubs:IntelDiscipline">
        <sch:assert test="//@intdis:CESVersion" flag="error">[PUBS-ID-00121][Error] If pubs:IntelDiscipline exists, then @intdis:CESVersion must exist as well.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.61 - ../Rules/PUBS_ID_00122.sch

Rule Description

[PUBS-ID-00122][Error] For element ddms:language, attribute ddms:qualifier must have a value in CVEnumIRMCompoundLanguageQualifierType.xml. Human Readable: If a qualifier is specified for a language, it must appear in the CompoundLanguageQualifierType CVE.

Code Description

This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter \$searchTerm is contained in the parameter \$list. The parameter \$searchTerm is relative in scope to the parameter \$context. The value for the parameter \$list is a variable defined in the main document (PUBS_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern id="PUBS-ID-00122" is-a="ValidateValueExistenceInList">
  <sch:param name="ruleText"
    value="' [PUBS-ID-00122][Error] For element ddms:language, attribute ddms:qualifier must have a value in CVEnumIRMCompoundLanguageQualifierType.xml. '"/>
  <sch:param name="codeDesc"
    value="' This rule uses an abstract pattern to consolidate logic. It checks that the value in parameter $searchTerm is contained in the parameter $list. The
parameter $searchTerm is relative in scope to the parameter $context. The value for the parameter $list is a variable defined in the main document (PUBS_XML.sch), which reads values from a
specific CVE file. '"/>
  <sch:param name="context" value="pubs:OriginalLanguageVersion//ddms:language"/>
  <sch:param name="searchTerm" value="@ddms:qualifier"/>
  <sch:param name="list" value="$compoundLanguageQualifierTypeList"/>
  <sch:param name="errMsg"
    value="' [PUBS-ID-00122][Error] For element ddms:language, attribute ddms:qualifier must have a value in CVEnumIRMCompoundLanguageQualifierType.xml. '"/>
</sch:pattern>
```

2.62 - ../Rules/PUBS_ID_00123.sch

Rule Description

[PUBS-ID-00123][Error] The @irm:DESVersion is less than the minimum version allowed: 12. Human Readable: The IRM version imported by PUBS must be greater than or equal to 12.

Code Description

For all elements that contain @irm:DESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 12.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00123">
  <sch:rule context="*[@irm:DESVersion]">
    <sch:let name="version"
      value="number(if (contains(@irm:DESVersion,'-')) then substring-before(@irm:DESVersion,'-') else @irm:DESVersion)"/>
    <sch:assert test="$version >= 12" flag="error">[PUBS-ID-00123][Error] The @irm:DESVersion is less than the minimum version allowed: 12. Human Readable: The IRM
version imported by PUBS must be greater than or equal to 12.</sch:assert>
  </sch:rule>
</sch:pattern>
```


2.63 - ../Rules/PUBS_ID_00124.sch

Rule Description

[PUBS-ID-00124][Error] The @src:DESVersion is less than the minimum version allowed: 201508. Human Readable: The SRC version imported by PUBS must be greater than or equal to 2015-AUG.

Code Description

For all elements that contain @src:DESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 201508.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00124">
  <sch:rule context="*[@src:DESVersion]">
    <sch:let name="version"
      value="number(if (contains(@src:DESVersion,'-')) then substring-before(@src:DESVersion,'-') else @src:DESVersion)"/>
    <sch:assert test="$version >= 201508" flag="error">[PUBS-ID-00124][Error] The @src:DESVersion is less than the minimum version allowed: 201508. Human Readable: The
SRC version imported by PUBS must be greater than or equal to 2015-AUG.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.64 - ../Rules/PUBS_ID_00125.sch

Rule Description

[PUBS-ID-00125][Error] The @tdf:version is less than the minimum version allowed: 3. Human Readable: The IC-TDF version imported by PUBS must be greater than or equal to 3.

Code Description

For all elements that contain @tdf:version, this rule verifies that the version is greater than or equal to the minimum allowed version: 3.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00125">
  <sch:rule context="*[@tdf:version]">
    <sch:let name="version"
      value="number(if (contains(@tdf:version,'-')) then substring-before(@tdf:version,'-') else @tdf:version)"/>
    <sch:assert test="$version >= 3" flag="error">[PUBS-ID-00125][Error] The @tdf:version is less than the minimum version allowed: 3. Human Readable: The IC-TDF version
imported by PUBS must be greater than or equal to 3.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.65 - ../Rules/PUBS_ID_00126.sch

Rule Description

[PUBS-ID-00126][Error] The @ism:ISMCATCESVersion is less than the minimum version allowed: 201505. Human Readable: The ISMCAT version imported by PUBS must be greater than or equal to 2015-MAY.

Code Description

For all elements that contain @ism:ISMCATCESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 201505.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00126">
    <sch:rule context="*[@ism:ISMCATCESVersion]">
        <sch:let name="version"
            value="number(if (contains(@ism:ISMCATCESVersion,'-')) then substring-before(@ism:ISMCATCESVersion,'-') else @ism:ISMCATCESVersion)"/>
        <sch:assert test="$version >= 201505" flag="error">[PUBS-ID-00126][Error] The @ism:ISMCATCESVersion is less than the minimum version allowed: 201505. Human Readable:
The ISMCAT version imported by PUBS must be greater than or equal to 2015-MAY.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.66 - ../Rules/PUBS_ID_00127.sch

Rule Description

[PUBS-ID-00127][Error] The @edh:DESVersion is less than the minimum version allowed: 4. Human Readable: The IC-EDH version imported by PUBS must be greater than or equal to 4.

Code Description

For all elements that contain @edh:DESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 4.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00127">
  <sch:rule context="*[@edh:DESVersion]">
    <sch:let name="version"
      value="number(if (contains(@edh:DESVersion,'-')) then substring-before(@edh:DESVersion,'-') else @edh:DESVersion)"/>
    <sch:assert test="$version >= 4" flag="error">[PUBS-ID-00127][Error] The @edh:DESVersion is less than the minimum version allowed: 4. Human Readable: The IC-EDH
version imported by PUBS must be greater than or equal to 4.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.67 - ../Rules/PUBS_ID_00128.sch

Rule Description

[PUBS-ID-00128][Error] The @virt:DESVersion is less than the minimum version allowed: 1. Human Readable: The VIRT version imported by PUBS must be greater than or equal to 1.

Code Description

For all descendant elements of pubs:IntelDoc that contain @virt:DESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 1.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00128">
  <sch:rule context="//pubs:IntelDoc//*[ @virt:DESVersion]">
    <sch:let name="version"
      value="number(if (contains(@virt:DESVersion,'-')) then substring-before(@virt:DESVersion,'-') else @virt:DESVersion)"/>
    <sch:assert test="$version >= 1" flag="error">[PUBS-ID-00128][Error] The @virt:DESVersion is less than the minimum version allowed: 1. Human Readable: The VIRT
version imported by PUBS must be greater than or equal to 1.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.68 - ../Rules/PUBS_ID_00129.sch

Rule Description

[PUBS-ID-00129][Error] If VIRT elements or attributes are used in PUBS, there must exist a descendant element of pubs:IntelDoc that has @virt:DESVersion. Human Readable: If elements or attributes of VIRT are used in PUBS, there must be a descendant element of pubs:IntelDoc that defines the VIRT DESVersion.

Code Description

If there exists a descendant element of pubs:IntelDoc that uses a VIRT element or attribute, there must exist a descendant element of pubs:IntelDoc that contains @virt:DESVersion.

Schematron Code

```
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="PUBS-ID-00129">
    <sch:rule context="//pubs:IntelDoc//virt:* | //pubs:IntelDoc//*[@virt:*)">
        <sch:assert test="exists(//pubs:IntelDoc//*[@virt:DESVersion])" flag="error">[PUBS-ID-00129][Error] If VIRT elements or attributes are used in PUBS, there must exist a
descendant element of pubs:IntelDoc that has @virt:DESVersion. Human Readable: If elements or attributes of VIRT are used in PUBS, there must be a descendant element of pubs:IntelDoc that
defines the VIRT DESVersion.</sch:assert>
    </sch:rule>
</sch:pattern>
```

Chapter 3 - Abstract Patterns

All of the Abstract Patterns for PUBS are listed in this section. These patterns may depend on variables defined in the Schematron Schema section.

3.1 - ./Lib/CompareDateTimes.sch

Code Description

This abstract pattern compares the date contained within the param \$context to each date contained within the param \$secondaryDateList (using the comparison operator contained in param \$operator) and makes sure that each comparison returns true.

Schematron Code

```
<?ICEA abstractPattern?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
--><!--
  $context := an xpath to an element containing a date
  $secondaryDateList := a list of dates in which the first entry is the date to compare to all other entries in the list
  $operator := the equality operator to use for comparing the dateTimes

  This abstract pattern compares the date contained within the param $context to each date
  contained within the param $secondaryDateList (using the comparison operator
  contained in param $operator) and makes sure that each comparison returns
  true.

-->
<sch:pattern abstract="true" id="CompareDateTimes">
  <sch:rule context="$context">
    <sch:assert test="if ($flag = 'warning') then every $secondaryDate in $secondaryDateList satisfies dtf:compareDateTimeRanges(string(.), $operator,
string($secondaryDate)) else true()"
      flag="warning">
      <sch:value-of select="$ruleText"/>
    </sch:assert>
    <sch:assert test="if ($flag = 'error') then every $secondaryDate in $secondaryDateList satisfies dtf:compareDateTimeRanges(string(.), $operator, string($secondaryDate))
else true()"
      flag="error">
      <sch:value-of select="$ruleText"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```


3.2 - ./Lib/ValidateValueExistenceInList.sch

Code Description

This abstract pattern checks to see if an attribute of an element exists in a list. \$context := the context in which the searchValue exists. \$searchTerm := the value which should be verified is in the list. \$list := the list in which to search for the searchValue. \$errMsg := the error message text to display when the assertion fails.

Schematron Code

```
<?ICEA abstractPattern?>
<!-- Notices - Distribution Notice:
    This document has been approved for Public Release and is available for use without restriction.
--><!--
    This abstract pattern checks to see if an attribute of an element exists in a list.

    $context      := the context in which the searchValue exists
    $searchTerm   := the value which you want to verify is in the list
    $list         := the list in which to search for the searchValue
    $errMsg       := the error message text to display when the assertion fails

    Example usage:
    <sch:pattern is-a="ValidateAttributeValueExistenceInList" id="PUBS_ID_00063" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:param name="context" value="pubs:QuantityReference[@unitOfMeasureVocabulary='ISO4217-3']"/>
        <sch:param name="searchTerm" value="@unitOfMeasure"/>
        <sch:param name="list" value="$iso4217TrigraphList"/>
        <sch:param name="errMsg" value="'
            [PUBS-ID-00063][Error]
            If the element QuantityReference has the attribute unitOfMeasureVocabulary
            with a value of ISO4217-3 the value of attribute unitOfMeasure must be
            in CVEnumISO4217Trigraph.xml.
        '"/>
    </sch:pattern>

    Note: $iso4217TrigraphList is defined in the main document, PUBS_XML.xml.
-->
<sch:pattern abstract="true" id="ValidateValueExistenceInList">
    <sch:rule context="$context">
        <sch:assert test="some $token in $list satisfies compare($token,normalize-space($searchTerm)) = 0"
            flag="error">
            <sch:value-of select="$errMsg"/>
        </sch:assert>
    </sch:rule>
</sch:pattern>
```

Chapter 4 - Schematron Schema

The top level Schematron file for PUBS is in this section. This file imports all of the others and also defines many global variables they are all dependent on.

4.1 - ./PUBS_XML.sch

Schematron Code

```
<?ICEA master?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:schema queryBinding="xslt2"><!-- ***** --><!-- * Namespace declarations * --><!-- ***** -->
<sch:ns uri="urn:us:gov:ic:pubs" prefix="pubs"/>
    <sch:ns uri="urn:us:gov:ic:icgenc" prefix="genc"/>
    <sch:ns uri="urn:us:gov:ic:edh" prefix="edh"/>
    <sch:ns uri="urn:us:gov:ic:src" prefix="src"/>
    <sch:ns uri="urn:us:gov:ic:cve" prefix="cve"/>
    <sch:ns uri="http://www.w3.org/1999/xlink" prefix="xlink"/>
    <sch:ns uri="urn:us:gov:ic:ism" prefix="ism"/>
    <sch:ns uri="urn:us:gov:ic:irm" prefix="irm"/>
    <sch:ns uri="urn:us:gov:ic:tdf" prefix="tdf"/>
    <sch:ns uri="urn:us:mil:ces:metadata:ddms:5" prefix="ddms"/>
    <sch:ns uri="http://www.w3.org/2001/XMLSchema" prefix="xs"/>
    <sch:ns uri="http://www.w3.org/1999/XSL/Transform" prefix="xsl"/>
    <sch:ns uri="date:time:function" prefix="dtf"/>
    <sch:ns uri="urn:us:gov:ic:intdis" prefix="intdis"/>
    <sch:ns uri="urn:us:gov:ic:mime" prefix="mime"/>
    <sch:ns uri="urn:us:gov:ic:virt" prefix="virt"/>
    <!-- ***** --><!-- * General Global Variables * --><!-- ***** -->
<sch:let name="currentYear" value="year-from-dateTime(current-dateTime())"/>
    <sch:let name="timeZoneRegEx" value="'Z|[\+-]\d{2}:\d{2}'"/>
    <sch:let name="endsWithTimeZoneRegEx" value="concat('^.*',$timeZoneRegEx,$')"/>
    <sch:let name="startDateTimeTemplate" value="'0001-01-01T00:00:00.000'"/>
    <sch:let name="endDateTimeTemplate" value="'9999-12-01T23:59:59.999'"/>
    <sch:let name="defaultTimeZone" value="'Z'"/>
    <!-- ***** --><!-- * List Definitions * --><!-- ***** -->
<sch:let name="coverageIso3166DigraphList"
    value="document('.../CVE/PUBS/CVEnumPubsCoverageISO3166Digraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="countryIso3166TrigraphList"
    value="document('.../CVE/PUBS/CVEnumPubsCountryISO3166Trigraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="iso639-2TrigraphList"
    value="document('.../CVE/PUBS/CVEnumISO639-2Trigraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="iso639-3TrigraphList"
    value="document('.../CVE/PUBS/CVEnumISO639-3Trigraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="iso639DigraphList"
    value="document('.../CVE/PUBS/CVEnumISO639Digraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="unce20UnitsOfMeasureList"
    value="document('.../CVE/PUBS/CVEnumUNCE20UnitsOfMeasure.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="iso4217TrigraphList"
    value="document('.../CVE/PUBS/CVEnumISO4217Trigraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="iso4217NumericList"
    value="document('.../CVE/PUBS/CVEnumISO4217Numeric.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="compoundLanguageQualifierTypeList"
    value="document('.../CVE/IRM/CVEnumIRMCompoundLanguageQualifierType.xml')//cve:Value"/>
    <!-- ***** --><!-- * Abstract Rule and Pattern Includes * --><!-- ***** -->
<sch:include href="./Lib/ValidateValueExistenceInList.sch"/>
```

```

    <sch:include href="./Lib/CompareDateTimes.sch"/>
    <sch:include href="./Lib/DateYearRangeRule.sch"/>
    <sch:include href="./Lib/DateListYearRangeRule.sch"/>
    <!-- ***** --><!-- * Custom XSLT2 Function Definitions * --><!-- ***** --><!--
Returns the maximum day of the month for an xs:dateTime as an xs:string.
@param {xs:dateTime} date The date time from which to get the month
@returns {xs:string} String representation of the maximum day of the month
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    name="dtf:getMaxDay"
    as="xs:string">
    <xsl:param name="date" as="xs:dateTime"/>
    <xsl:variable name="month" select="number(dtf:getMonth(string($date)))"/>
    <xsl:choose>
        <xsl:when test="$month = (1,3,5,7,8,10,12)">
            <xsl:value-of select="31"/>
        </xsl:when>
        <xsl:when test="$month = (4,6,9,11)">
            <xsl:value-of select="30"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:choose>
                <xsl:when test="dtf:isLeapYear(string($date))">
                    <xsl:value-of select="29"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:value-of select="28"/>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:otherwise>
    </xsl:choose>
</xsl:function>
<!--
@param {xs:date} date String representation of a date
@returns {xs:boolean} Returns true if the date provided occurs in a
    leap year; otherwise returns false.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    name="dtf:isLeapYear"
    as="xs:boolean">
    <xsl:param name="date" as="xs:string"/>
    <xsl:variable name="year" as="xs:integer" select="xs:integer(dtf:getYear($date))"/>
    <xsl:choose>
        <xsl:when test="$year mod 100 = 0">
            <xsl:choose>
                <xsl:when test="$year mod 400 = 0">
                    <xsl:value-of select="true()"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:value-of select="false()"/>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:when>
        <xsl:otherwise>

```

```

        <xsl:choose>
          <xsl:when test="$year mod 4 = 0">
            <xsl:value-of select="true()"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select="false()"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:function>
  <!--
  Replaces the day portion of the provided dateTime with the new day provided.
  @param {xs:dateTime} dateTime An xs:dateTime to be updated with new day.
  @param {xs:string} newDayString String representation of day portion of a date.
  @returns {xs:dateTime} Returns new xs:dateTime with updated day portion.
  leap year; otherwise returns false.
  -->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:replaceDateTimeDay"
              as="xs:dateTime">
  <xsl:param name="dateTime" as="xs:dateTime"/>
  <xsl:param name="newDayString" as="xs:string"/>
  <xsl:variable name="beforeDay" select="substring(string($dateTime), 1, 8)"/>
  <xsl:variable name="afterDay" select="substring(string($dateTime), 11)"/>
  <xsl:value-of select="concat($beforeDay, $newDayString, $afterDay)"/>
</xsl:function>
  <!--
  Returns a string representation of the year portion of the date
  represented by the provided string.
  @param {xs:string} dateString String representation of a date in one
    of the allowable formats.
  @returns {xs:string} String representation of the year portion of the
    date represented by the provided string.
  -->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:getYear"
              as="xs:string">
  <xsl:param name="dateString" as="xs:string"/>
  <xsl:value-of select="substring(dtf:removeTimeZone($dateString), 1, 4)"/>
</xsl:function>
  <!--
  Returns a string representation of the month portion of the date
  represented by the provided string.
  @param {xs:string} dateString String representation of a date in one
    of the allowable formats.
  @returns {xs:string} String representation of the month portion of the
    date represented by the provided string.
  -->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:getMonth"
              as="xs:string">
  <xsl:param name="dateString" as="xs:string"/>
  <xsl:value-of select="substring(dtf:removeTimeZone($dateString), 6, 2)"/>

```

```

        </xsl:function>
        <!--
        Returns a string representation of the day portion of the date
        represented by the provided string.
        @param {xs:string} dateString String representation of a date in one
        of the allowable formats.
        @returns {xs:string} String representation of the day portion of the
        date represented by the provided string.
        -->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:getDay"
              as="xs:string">
    <xsl:param name="dateString" as="xs:string"/>
    <xsl:value-of select="substring(dtf:removeTimeZone($dateString), 9, 2)"/>
</xsl:function>
<!--
Returns a string representation of the timezone portion of the date
represented by the provided string.
@param {xs:string} dateString String representation of a date in one
of the allowable formats.
@returns {xs:string} String representation of the timezone portion of
the date represented by the provided string.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:getTimeZone"
              as="xs:string">
    <xsl:param name="dateString" as="xs:string"/>
    <xsl:variable name="dateTimeEndingWithTimezone"
                  as="xs:string"
                  select="concat('^.*(', $timezoneRegex, ')$')"/>
    <xsl:choose>
        <xsl:when test="matches($dateString, $dateTimeEndingWithTimezone)">
            <xsl:value-of select="replace($dateString, $dateTimeEndingWithTimezone, '$1')"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="$defaultTimeZone"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:function>
<!--
Returns true if the year portion of the date represented by the provided
string contains four (4) digits; otherwise returns false.
@param {xs:string} dateString String representation of a date in one
of the allowable formats.
@returns {xs:string} true if the year portion of the date represented by
the provided string contains four (4) digits; otherwise returns false.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:yearPortionHasFourDigits"
              as="xs:boolean">
    <xsl:param name="dateString" as="xs:string"/>
    <xsl:variable name="dateWithOnlyFourDigitYearAndOptionalTimeZoneRegex"
                  as="xs:string"
                  select="concat('^\\d{4}(', $timezoneRegex, ')?$')"/>

```

```

        <xsl:variable name="dateStartingWithFourDigitYearRegEx"
                    as="xs:string"
                    select="'^\\d{4}-.*$'"/>
    <xsl:value-of select="matches($dateString, $dateWithOnlyFourDigitYearAndOptionalTimeZoneRegEx) or matches($dateString, $dateStartingWithFourDigitYearRegEx)"/>
</xsl:function>
<!--
Removes the timezone portion of the date represented by the provided
string and returns all remaining portions.
@param {xs:string} dateString String representation of a date in one
of the allowable formats.
@returns {xs:string} String representation of a date without a timezone
portion.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:removeTimeZone"
              as="xs:string">
    <xsl:param name="dateString" as="xs:string"/>
    <xsl:value-of select="replace($dateString, $timeZoneRegEx, '')"/>
</xsl:function>
<!--
Uses the template provided to fill in missing portions of the string
representation of a dateTime provided and returns a full xs:dateTime.
The dateString provided must not contain a timezone.
@param {xs:string} dateString String representation of a date in one
of the allowable formats.
@param {xs:string} dateTemplateString String template of a default date
from which to pad missing portions of the dateString parameter.
@returns {xs:dateTime} An xs:dateTime represented by the string date provided.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:padDateTimeWithTemplate"
              as="xs:dateTime">
    <xsl:param name="dateString" as="xs:string"/>
    <xsl:param name="dateTemplateString" as="xs:string"/>
    <xsl:value-of select="concat($dateString, substring($dateTemplateString, string-length(normalize-space($dateString))+1))"/>
</xsl:function>
<!--
Returns true if the string provided represents an allowable dateTime
format; false, otherwise. The allowable dateTime formats are defined
in the DES for the PUBS.XML specification.
@returns {xs:boolean} Returns true if the string provided represents an
allowable dateTime format; false, otherwise.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:isAllowableDateTimeFormat"
              as="xs:boolean">
    <xsl:param name="input" as="xs:string"/>
    <xsl:variable name="trimmedInput" as="xs:string" select="normalize-space($input)"/>
    <!-- year -->
    <xsl:variable name="YYYY" as="xs:string" select="'^\\d{4}(Z|\\[\\+-]\\d{2}:\\d{2})?$'"/>
    <!-- year, month -->
    <xsl:variable name="YYYY-MM"
                  as="xs:string"
                  select="'^\\d{4}-\\d{2}(Z|\\[\\+-]\\d{2}:\\d{2})?$'"/>

```

```

        <!-- year, month, day -->
<xsl:variable name="YYYY-MM-DD"
              as="xs:string"
              select="'^\\d{4}-\\d{2}-\\d{2}(Z|[\\+-]\\d{2}:\\d{2})?\\$'"/>
        <!-- year, month, day, hour, minute -->
<xsl:variable name="YYYY-MM-DDThh-mm"
              as="xs:string"
              select="'^\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}(Z|[\\+-]\\d{2}:\\d{2})?\\$'"/>
        <!-- year, month, day, hour, minute, seconds, optional milliseconds -->
<xsl:variable name="YYYY-MM-DDThh-mm-ss"
              as="xs:string"
              select="'^\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}(\\.\\d{1,})?(Z|[\\+-]\\d{2}:\\d{2})?\\$'"/>
        <xsl:value-of select="matches($trimmedInput, $YYYY) or matches($trimmedInput, $YYYY-MM) or matches($trimmedInput, $YYYY-MM-DD) or matches($trimmedInput, $YYYY-MM-DDThh-mm) or matches($trimmedInput, $YYYY-MM-DDThh-mm-ss)"/>
    </xsl:function>
    <!--
    Returns the earliest xs:dateTime possible for the provided string
    representation of a dateTime. Fills in missing portions of the
    dateTime with the earliest possible values. Default values for missing
    portions:
    MM = 01
    DD = 01
    hh = 00
    mm = 00
    ss = 00
    s  = 000
    @param {xs:string} dateString String representation of a date in one
    of the allowable formats.
    @returns {xs:dateTime} The earliest xs:dateTime possible for the
    provided string representation of a dateTime.
    -->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:startDate"
              as="xs:dateTime">
    <xsl:param name="dateString" as="xs:string"/>
    <xsl:variable name="timeZonePortion" select="dtf:getTimeZone($dateString)"/>
    <xsl:variable name="dateTimePortion" select="dtf:removeTimeZone($dateString)"/>
    <xsl:variable name="outputDate"
                  select="dtf:padDateTimeWithTemplate($dateTimePortion, $startDateTimeTemplate)"/>
    <xsl:value-of select="concat($outputDate, $timeZonePortion)"/>
</xsl:function>
    <!--
    Returns the latest xs:dateTime possible for the provided string
    representation of a dateTime. Fills in missing portions of the
    dateTime with the latest possible values. Default values for missing
    portions:
    MM = 12
    DD = maximum day of the month
    hh = 23
    mm = 59
    ss = 59
    s  = 999
    @param {xs:string} dateString String representation of a date in one
    of the allowable formats.
    -->
```



```

    @returns {xs:dateTime} The latest xs:dateTime possible for the
        provided string representation of a dateTime.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    name="dtf:endDate"
    as="xs:dateTime">
    <xsl:param name="input" as="xs:string"/>
    <xsl:variable name="timeZonePortion" select="dtf:getTimeZone($input)"/>
    <xsl:variable name="dateTimePortion" select="dtf:removeTimeZone($input)"/>
    <xsl:variable name="outputDate"
        select="dtf:padDateTimeWithTemplate($dateTimePortion, $endDateTimeTemplate)"/>
    <xsl:variable name="outputWithCorrectedDay"
        select="dtf:replaceDateTimeDay($outputDate, dtf:getMaxDay($outputDate))"/>
    <xsl:choose>
        <xsl:when test="dtf:getDay($input)">
            <xsl:value-of select="concat($outputDate, $timeZonePortion)"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="concat($outputWithCorrectedDay, $timeZonePortion)"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:function>
<!--
Calculates the date range implied for both primary and secondary and
determines if there is any overlap between the two ranges. Overlap is
defined as the start of primary date range less than or equal to the
end of secondary date range, inclusive, and the start of the secondary
date range less than or equal to the end of the primary date range.
Returns true if there is any overlap; otherwise, returns false.
@param {xs:string} primary String representation of a date in one
    of the allowable formats.
@param {xs:string} secondary String representation of a date in one
    of the allowable formats.
@returns {xs:boolean} Returns true if the date ranges implied by primary
    and secondary overlap at all; otherwise, returns false.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    name="dtf:overlaps"
    as="xs:boolean">
    <xsl:param name="primary" as="xs:string"/>
    <xsl:param name="secondary" as="xs:string"/>
    <xsl:variable name="primaryStart"
        as="xs:dateTime"
        select="dtf:startDate($primary)"/>
    <xsl:variable name="primaryEnd" as="xs:dateTime" select="dtf:endDate($primary)"/>
    <xsl:variable name="secondaryStart"
        as="xs:dateTime"
        select="dtf:startDate($secondary)"/>
    <xsl:variable name="secondaryEnd"
        as="xs:dateTime"
        select="dtf:endDate($secondary)"/>
    <xsl:value-of select="$primaryStart <= $secondaryEnd and $secondaryStart <= $primaryEnd"/>
</xsl:function>
<!--

```

```

    Determines if the date range implied by the string representation in
    primary is stricly before the date range implied by the string
    representation in secondary. Returns true if the end of the date
    range implied by primary is less than the start of the date range
    implied by secondary; otherwise, returns false.
    @param {xs:string} primary String representation of a date in one
        of the allowable formats.
    @param {xs:string} secondary String representation of a date in one
        of the allowable formats.
    @returns {xs:boolean} Returns true if the date range implied by primary
        is stricly earlier than the date range implied by secondary; otherwise,
        returns false.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:isBefore"
              as="xs:boolean">
  <xsl:param name="primary" as="xs:string"/>
  <xsl:param name="secondary" as="xs:string"/>
  <xsl:variable name="primaryEnd" as="xs:dateTime" select="dtf:endDate($primary)"/>
  <xsl:variable name="secondaryStart"
                as="xs:dateTime"
                select="dtf:startDate($secondary)"/>
  <xsl:value-of select="$primaryEnd < $secondaryStart"/>
</xsl:function>
<!--
    Determines if the date range implied by the string representation in
    primary is stricly after the date range implied by the string
    representation in secondary. Returns true if the end of the date
    range implied by primary is less than the start of the date range
    implied by secondary; otherwise, returns false.
    @param {xs:string} primary String representation of a date in one
        of the allowable formats.
    @param {xs:string} secondary String representation of a date in one
        of the allowable formats.
    @returns {xs:boolean} Returns true if the date range implied by primary
        is stricly later than the date range implied by secondary; otherwise,
        returns false.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:isAfter"
              as="xs:boolean">
  <xsl:param name="primary" as="xs:string"/>
  <xsl:param name="secondary" as="xs:string"/>
  <xsl:variable name="primaryStart"
                as="xs:dateTime"
                select="dtf:startDate($primary)"/>
  <xsl:variable name="secondaryEnd"
                as="xs:dateTime"
                select="dtf:endDate($secondary)"/>
  <xsl:value-of select="$secondaryEnd < $primaryStart"/>
</xsl:function>
<!--
    Determines if the date range implied by the string representation in
    primary satisfies the comparison to the date range implied by secondary

```

using the provided comparison operator; otherwise, returns false.

Both primary and secondary must be in one of the allowable formats and represent dates with four digits in the year portion.

@param {xs:string} primary String representation of a date in one of the allowable formats.

@param {xs:string} secondary String representation of a date in one of the allowable formats.

@returns {xs:boolean} Returns true if the date range implied by primary satisfies the comparison to the date range implied by secondary using the provided comparison operator; otherwise, returns false.

-->

```
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:compareDateTimeRanges"
              as="xs:boolean">
  <xsl:param name="primary" as="xs:string"/>
  <xsl:param name="operator" as="xs:string"/>
  <xsl:param name="secondary" as="xs:string"/>
  <xsl:variable name="primaryAndSecondaryYearPortionsHaveFourDigits"
              as="xs:boolean"
              select="dtf:yearPortionHasFourDigits($primary) and dtf:yearPortionHasFourDigits($secondary)"/>
  <xsl:choose>
    <xsl:when test="$primaryAndSecondaryYearPortionsHaveFourDigits">
      <xsl:variable name="primaryStart"
                  as="xs:dateTime"
                  select="dtf:startDate($primary)"/>
      <xsl:variable name="primaryEnd" as="xs:dateTime" select="dtf:endDate($primary)"/>
      <xsl:variable name="secondaryStart"
                  as="xs:dateTime"
                  select="dtf:startDate($secondary)"/>
      <xsl:variable name="secondaryEnd"
                  as="xs:dateTime"
                  select="dtf:endDate($secondary)"/>
      <xsl:choose><!-- 'Less Than' Edge Case --><!-- 2010-01-01T00:00:00.000Z < 2010 -->
        <xsl:when test="($operator = 'lt' or $operator = '&lt;') and (($primaryStart = $primaryEnd and $primaryStart = $secondaryStart) or ($primaryStart = $primaryEnd and $primaryStart = $secondaryEnd) or ($secondaryStart = $secondaryEnd and $primaryStart = $secondaryStart))">
          <xsl:value-of select="false()"/>
        </xsl:when>
        <!-- 'Greater Than' Edge Case --><!-- 2010-12-31T23:59:59.999Z > 2010 -->
        <xsl:when test="($operator = 'gt' or $operator = '&gt;') and (($primaryStart = $primaryEnd and $primaryEnd = $secondaryEnd) or ($primaryStart = $primaryEnd and $primaryEnd = $secondaryStart) or ($secondaryStart = $secondaryEnd and $primaryEnd = $secondaryEnd))">
          <xsl:value-of select="false()"/>
        </xsl:when>
        <!-- 'Less Than' and 'Less Than or Equal' -->
        <xsl:when test="$operator = 'lt' or $operator = '&lt;' or $operator = '&lt;='">
          <xsl:value-of select="dtf:isBefore($primary, $secondary) or dtf:overlaps($primary, $secondary)"/>
        </xsl:when>
        <!-- 'Greater Than' and 'Greater Than or Equal' -->
        <xsl:when test="$operator = 'gt' or $operator = '&gt;' or $operator = '&gt;='">
          <xsl:value-of select="dtf:isAfter($primary, $secondary) or dtf:overlaps($primary, $secondary)"/>
        </xsl:when>
        <!-- Default to false -->
      <xsl:otherwise>
        <xsl:value-of select="false()"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:choose>
</xsl:function>
```

```
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="false()" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:function>
<!--*****--><!-- (U) PUBS ID Rules --><!--*****--><!--(U) -->
<sch:include href="./Rules/PUBS_ID_00001.sch"/>
<sch:include href="./Rules/PUBS_ID_00003.sch"/>
<sch:include href="./Rules/PUBS_ID_00008.sch"/>
<sch:include href="./Rules/PUBS_ID_00009.sch"/>
<sch:include href="./Rules/PUBS_ID_00017.sch"/>
<sch:include href="./Rules/PUBS_ID_00027.sch"/>
<sch:include href="./Rules/PUBS_ID_00028.sch"/>
<sch:include href="./Rules/PUBS_ID_00029.sch"/>
<sch:include href="./Rules/PUBS_ID_00030.sch"/>
<sch:include href="./Rules/PUBS_ID_00031.sch"/>
<sch:include href="./Rules/PUBS_ID_00032.sch"/>
<sch:include href="./Rules/PUBS_ID_00033.sch"/>
<sch:include href="./Rules/PUBS_ID_00034.sch"/>
<sch:include href="./Rules/PUBS_ID_00035.sch"/>
<sch:include href="./Rules/PUBS_ID_00036.sch"/>
<sch:include href="./Rules/PUBS_ID_00040.sch"/>
<sch:include href="./Rules/PUBS_ID_00042.sch"/>
<sch:include href="./Rules/PUBS_ID_00044.sch"/>
<sch:include href="./Rules/PUBS_ID_00045.sch"/>
<sch:include href="./Rules/PUBS_ID_00046.sch"/>
<sch:include href="./Rules/PUBS_ID_00047.sch"/>
<sch:include href="./Rules/PUBS_ID_00048.sch"/>
<sch:include href="./Rules/PUBS_ID_00049.sch"/>
<sch:include href="./Rules/PUBS_ID_00050.sch"/>
<sch:include href="./Rules/PUBS_ID_00053.sch"/>
<sch:include href="./Rules/PUBS_ID_00054.sch"/>
<sch:include href="./Rules/PUBS_ID_00055.sch"/>
<sch:include href="./Rules/PUBS_ID_00062.sch"/>
<sch:include href="./Rules/PUBS_ID_00063.sch"/>
<sch:include href="./Rules/PUBS_ID_00064.sch"/>
<sch:include href="./Rules/PUBS_ID_00065.sch"/>
<sch:include href="./Rules/PUBS_ID_00070.sch"/>
<sch:include href="./Rules/PUBS_ID_00072.sch"/>
<sch:include href="./Rules/PUBS_ID_00079.sch"/>
<sch:include href="./Rules/PUBS_ID_00080.sch"/>
<sch:include href="./Rules/PUBS_ID_00081.sch"/>
<sch:include href="./Rules/PUBS_ID_00082.sch"/>
<sch:include href="./Rules/PUBS_ID_00086.sch"/>
<sch:include href="./Rules/PUBS_ID_00087.sch"/>
<sch:include href="./Rules/PUBS_ID_00092.sch"/>
<sch:include href="./Rules/PUBS_ID_00093.sch"/>
<sch:include href="./Rules/PUBS_ID_00096.sch"/>
<sch:include href="./Rules/PUBS_ID_00097.sch"/>
<sch:include href="./Rules/PUBS_ID_00098.sch"/>
<sch:include href="./Rules/PUBS_ID_00099.sch"/>
```

```
<sch:include href="./Rules/PUBS_ID_00100.sch"/>
<sch:include href="./Rules/PUBS_ID_00101.sch"/>
<sch:include href="./Rules/PUBS_ID_00103.sch"/>
<sch:include href="./Rules/PUBS_ID_00104.sch"/>
<sch:include href="./Rules/PUBS_ID_00110.sch"/>
<sch:include href="./Rules/PUBS_ID_00111.sch"/>
<sch:include href="./Rules/PUBS_ID_00113.sch"/>
<sch:include href="./Rules/PUBS_ID_00114.sch"/>
<sch:include href="./Rules/PUBS_ID_00115.sch"/>
<sch:include href="./Rules/PUBS_ID_00116.sch"/>
<sch:include href="./Rules/PUBS_ID_00117.sch"/>
<sch:include href="./Rules/PUBS_ID_00118.sch"/>
<sch:include href="./Rules/PUBS_ID_00119.sch"/>
<sch:include href="./Rules/PUBS_ID_00120.sch"/>
<sch:include href="./Rules/PUBS_ID_00121.sch"/>
<sch:include href="./Rules/PUBS_ID_00122.sch"/>
<sch:include href="./Rules/PUBS_ID_00123.sch"/>
<sch:include href="./Rules/PUBS_ID_00124.sch"/>
<sch:include href="./Rules/PUBS_ID_00125.sch"/>
<sch:include href="./Rules/PUBS_ID_00126.sch"/>
<sch:include href="./Rules/PUBS_ID_00127.sch"/>
<sch:include href="./Rules/PUBS_ID_00128.sch"/>
<sch:include href="./Rules/PUBS_ID_00129.sch"/>
</sch:schema>
```

Chapter 5 - Removed Rules

All of the numbered Rules for PUBS that have been removed are listed in this section. This section is just a reference for what rule numbers have been dropped. In many but not all cases there will be a reason listed. In all cases the version that the rule was dropped in is listed.

5.1 - `./Rules/deleted/PUBS_ID_00002.sch`

Rule Description

[PUBS-ID-00002] Removed in V6. Replaced by PUBS-ID-00093.

5.2 - `./Rules/deleted/PUBS_ID_00004.sch`

Rule Description

[PUBS-ID-00004] Removed in V4.

5.3 - `./Rules/deleted/PUBS_ID_00005.sch`

Rule Description

[PUBS-ID-00005] Removed in V2. Replaced by PUBS-ID-00065.

5.4 - `./Rules/deleted/PUBS_ID_00006.sch`

Rule Description

[PUBS-ID-00006] Removed in V2. Replaced by PUBS-ID-00065.

5.5 - `./Rules/deleted/PUBS_ID_00007.sch`

Rule Description

[PUBS-ID-00007] Removed in V10.

5.6 - `./Rules/deleted/PUBS_ID_00010.sch`

Rule Description

[PUBS-ID-00010] Removed in V2. Replaced by PUBS_ID_00066, PUBS_ID_00067, PUBS_ID_00068, and PUBS_ID_00069.

5.7 - `./Rules/deleted/PUBS_ID_00011.sch`

Rule Description

[PUBS-ID-00011] Removed in V5.

5.8 - **./Rules/deleted/PUBS_ID_00012.sch**

Rule Description

[PUBS-ID-00012] Removed in V2. Replaced by PUBS-ID-00071, PUBS_ID_00072, and PUBS_ID_00073.

5.9 - **./Rules/deleted/PUBS_ID_00013.sch**

Rule Description

[PUBS-ID-00013] Removed in V2. Replaced by PUBS-ID-00056, PUBS_ID_00057, PUBS_ID_00058, PUBS_ID_00059, PUBS_ID_00060, and PUBS_ID_00061.

5.10 - **./Rules/deleted/PUBS_ID_00014.sch**

Rule Description

[PUBS-ID-00014] Removed in V2. Replaced by PUBS-ID-00062, PUBS_ID_00063, and PUBS_ID_00064.

5.11 - **./Rules/deleted/PUBS_ID_00015.sch**

Rule Description

[PUBS-ID-00015][Error] Removed in V10. Covered by IRM_ID_00029.

5.12 - **./Rules/deleted/PUBS_ID_00016.sch**

Rule Description

[PUBS-ID-00016] Removed in V6. Replaced by PUBS-ID-00093.

5.13 - **./Rules/deleted/PUBS_ID_00018.sch**

Rule Description

[PUBS-ID-00018] Removed in V10.

5.14 - **./Rules/deleted/PUBS_ID_00019.sch**

Rule Description

[PUBS-ID-00019] Removed in V10.

5.15 - **./Rules/deleted/PUBS_ID_00020.sch**

Rule Description

[PUBS-ID-00020][Error] Removed in 2015-AUG.

5.16 - **./Rules/deleted/PUBS_ID_00021.sch**

Rule Description

[PUBS-ID-00021][Error] Removed in v2015-AUG.

5.17 - **./Rules/deleted/PUBS_ID_00022.sch**

Rule Description

[PUBS-ID-00022][Error] Removed in V11. Covered by IRM_ID_00073.

5.18 - **./Rules/deleted/PUBS_ID_00023.sch**

Rule Description

[PUBS-ID-00023][Error] Removed in V11. Covered by IRM_ID_00073.

5.19 - **./Rules/deleted/PUBS_ID_00024.sch**

Rule Description

[PUBS-ID-00024][Error]Removed in 2016-SEP since otherVocabulary was removed.

5.20 - **./Rules/deleted/PUBS_ID_00025.sch**

Rule Description

[PUBS-ID-00025][Error] Removed in 2016-SEP since otherVocabulary was removed.

5.21 - **./Rules/deleted/PUBS_ID_00026.sch**

Rule Description

[PUBS-ID-00026][Error] Removed in PUBS V10. Covered by IRM rules handling ID requirements.

5.22 - **./Rules/deleted/PUBS_ID_00037.sch**

Rule Description

[PUBS-ID-00037][Error] Rule removed in V11 because the otherNetwork attribute was removed. Networks Names are now controlled through the VIRT CVE, which uses a regex for other network names.

5.23 - **./Rules/deleted/PUBS_ID_00038.sch**

Rule Description

[PUBS-ID-00038][Error] Rule removed in V11 because the otherNetwork attribute was removed. Networks Names are now controlled through the VIRT CVE, which uses a regex for other network names.

5.24 - **./Rules/deleted/PUBS_ID_00039.sch**

Rule Description

[PUBS-ID-00039][Error] Rule removed in V11 because the otherNetwork attribute was removed. Networks Names are now controlled through the VIRT CVE, which uses a regex for other network names.

5.25 - **./Rules/deleted/PUBS_ID_00041.sch**

Rule Description

[PUBS-ID-00041][Error] Removed in version 11 for duplication with IRM rule.

5.26 - **./Rules/deleted/PUBS_ID_00043.sch**

Rule Description

[PUBS-ID-00043][Error] Rule removed in V9

5.27 - **./Rules/deleted/PUBS_ID_00051.sch**

Rule Description

[PUBS-ID-00051][] Removed in V5.

5.28 - **./Rules/deleted/PUBS_ID_00052.sch**

Rule Description

[PUBS-ID-00052][] Removed in V5.

5.29 - ~~./Rules/deleted/PUBS_ID_00056.sch~~

Rule Description

[PUBS-ID-00056][Error] Removed in v11. Duplicate of IRM rule.

5.30 - ~~./Rules/deleted/PUBS_ID_00057.sch~~

Rule Description

[PUBS-ID-00057][Error] Removed in v11. Duplicate of IRM rule.

5.31 - ~~./Rules/deleted/PUBS_ID_00058.sch~~

Rule Description

[PUBS-ID-00058][Error] Removed in v11. Duplicate of IRM rule.

5.32 - ~~./Rules/deleted/PUBS_ID_00059.sch~~

Rule Description

[PUBS-ID-00059][Error] Removed in v11. Duplicate of IRM rule.

5.33 - ~~./Rules/deleted/PUBS_ID_00060.sch~~

Rule Description

[PUBS-ID-00060][Error] Removed in v11. Duplicate of IRM rule.

5.34 - ~~./Rules/deleted/PUBS_ID_00061.sch~~

Rule Description

[PUBS-ID-00061][Error] Removed in v11. Duplicate of IRM rule.

5.35 - ~~./Rules/deleted/PUBS_ID_00066.sch~~

Rule Description

[PUBS-ID-00066][Error] Removed in PUBS V10. Covered by IRM_ID_00001.

5.36 - **./Rules/deleted/PUBS_ID_00067.sch**

Rule Description

[PUBS-ID-00067][] Removed in V6

5.37 - **./Rules/deleted/PUBS_ID_00068.sch**

Rule Description

[PUBS-ID-00068][Error] Removed in PUBSv10. Covered by IRM_ID_00003.

5.38 - **./Rules/deleted/PUBS_ID_00069.sch**

Rule Description

[PUBS-ID-00069][] Removed in V6.

5.39 - **./Rules/deleted/PUBS_ID_00071.sch**

Rule Description

[PUBS-ID-00071][Error] Rule removed in V11 because CVEnumPubsCountryFIPSDigraph.xml was no longer supported.

5.40 - **./Rules/deleted/PUBS_ID_00073.sch**

Rule Description

[PUBS-ID-00073][] Removed in V6.

5.41 - **./Rules/deleted/PUBS_ID_00074.sch**

Rule Description

[PUBS-ID-00074][Error] Removed in V11. Duplicates and IRM rule.

5.42 - **./Rules/deleted/PUBS_ID_00075.sch**

Rule Description

[PUBS-ID-00075][Error] Removed in v2016-SEP.

5.43 - **./Rules/deleted/PUBS_ID_00076.sch**

Rule Description

[PUBS-ID-00076][Error] Removed in v2015-AUG.

5.44 - **./Rules/deleted/PUBS_ID_00077.sch**

Rule Description

[PUBS-ID-00077][] Removed in V4.

5.45 - **./Rules/deleted/PUBS_ID_00078.sch**

Rule Description

[PUBS-ID-00078][] Removed in V4.

5.46 - **./Rules/deleted/PUBS_ID_00083.sch**

Rule Description

[PUBS-ID-00083][] Removed in V4.

5.47 - **./Rules/deleted/PUBS_ID_00084.sch**

Rule Description

[PUBS-ID-00084][] Introduced and removed in V4. Never made it to signed package.

5.48 - **./Rules/deleted/PUBS_ID_00085.sch**

Rule Description

[PUBS-ID-00085][] Removed in V6.

5.49 - **./Rules/deleted/PUBS_ID_00088.sch**

Rule Description

[PUBS-ID-00088][Error] Removed in V11. This rule is being covered under IRM-ID-00088.

5.50 - **./Rules/deleted/PUBS_ID_00089.sch**

Rule Description

[PUBS-ID-00089][Error] Removed in V11. Added to IRM

5.51 - **./Rules/deleted/PUBS_ID_00090.sch**

Rule Description

[PUBS-ID-00090][Error] Removed in V11. Covered under IRM_ID_00078.

5.52 - **./Rules/deleted/PUBS_ID_00091.sch**

Rule Description

[PUBS-ID-00091][Error] Removed in V10. Coverage moved to IRM

5.53 - **./Rules/deleted/PUBS_ID_00094.sch**

Rule Description

[PUBS-ID-00094][Error] Removed in PUBSv10. Rule covered by IC_IRM_00030.

5.54 - **./Rules/deleted/PUBS_ID_00102.sch**

Rule Description

[PUBS-ID-00102][] Removed in V11.

5.55 - **./Rules/deleted/PUBS_ID_00105.sch**

Rule Description

[PUBS-ID-00105][Error] Removed in V11. Rule is now covered by IRM-ID-00077.

5.56 - **./Rules/deleted/PUBS_ID_00106.sch**

Rule Description

[PUBS-ID-00106][Error] Removed in V11. This business rule will now be covered by the TSPI specification.

5.57 - ~~./Rules/deleted/PUBS_ID_00107.sch~~

Rule Description

[PUBS-ID-00107][Error] Removed in V11. This rule is now covered under IRM_ID_00076.

5.58 - ~~./Rules/deleted/PUBS_ID_00108.sch~~

Rule Description

[PUBS-ID-00108][Error] Removed in PUBSv10. Covered by IRM_ID_00037.

5.59 - ~~./Rules/deleted/PUBS_ID_00109.sch~~

Rule Description

[PUBS-ID-00109][Error] Removed in V11. Rule is now covered by IRM-ID-00074

5.60 - ~~./Rules/deleted/PUBS_ID_00112.sch~~

Rule Description

[PUBS-ID-00112][Error] Added and Removed in V11. Did not make it to final signature.