



Guide to Schematron Rules and Patterns

IRM 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/IRM_ID_00002.sch	3
2.2 - ../Rules/IRM_ID_00005.sch	4
2.3 - ../Rules/IRM_ID_00006.sch	5
2.4 - ../Rules/IRM_ID_00007.sch	6
2.5 - ../Rules/IRM_ID_00010.sch	7
2.6 - ../Rules/IRM_ID_00015.sch	8
2.7 - ../Rules/IRM_ID_00016.sch	9
2.8 - ../Rules/IRM_ID_00017.sch	10
2.9 - ../Rules/IRM_ID_00019.sch	11
2.10 - ../Rules/IRM_ID_00020.sch	12
2.11 - ../Rules/IRM_ID_00021.sch	13
2.12 - ../Rules/IRM_ID_00022.sch	14
2.13 - ../Rules/IRM_ID_00023.sch	15
2.14 - ../Rules/IRM_ID_00024.sch	16
2.15 - ../Rules/IRM_ID_00025.sch	17
2.16 - ../Rules/IRM_ID_00029.sch	18
2.17 - ../Rules/IRM_ID_00030.sch	19
2.18 - ../Rules/IRM_ID_00033.sch	20
2.19 - ../Rules/IRM_ID_00034.sch	21
2.20 - ../Rules/IRM_ID_00036.sch	22
2.21 - ../Rules/IRM_ID_00040.sch	23
2.22 - ../Rules/IRM_ID_00041.sch	24
2.23 - ../Rules/IRM_ID_00042.sch	25
2.24 - ../Rules/IRM_ID_00043.sch	26
2.25 - ../Rules/IRM_ID_00044.sch	27
2.26 - ../Rules/IRM_ID_00045.sch	28
2.27 - ../Rules/IRM_ID_00046.sch	29
2.28 - ../Rules/IRM_ID_00047.sch	30
2.29 - ../Rules/IRM_ID_00048.sch	31
2.30 - ../Rules/IRM_ID_00050.sch	32
2.31 - ../Rules/IRM_ID_00051.sch	33
2.32 - ../Rules/IRM_ID_00052.sch	34
2.33 - ../Rules/IRM_ID_00053.sch	35
2.34 - ../Rules/IRM_ID_00054.sch	36
2.35 - ../Rules/IRM_ID_00055.sch	37
2.36 - ../Rules/IRM_ID_00059.sch	38
2.37 - ../Rules/IRM_ID_00062.sch	39
2.38 - ../Rules/IRM_ID_00063.sch	40
2.39 - ../Rules/IRM_ID_00064.sch	41
2.40 - ../Rules/IRM_ID_00065.sch	42

2.41 - ../Rules/IRM_ID_00068.sch	43
2.42 - ../Rules/IRM_ID_00070.sch	44
2.43 - ../Rules/IRM_ID_00071.sch	45
2.44 - ../Rules/IRM_ID_00072.sch	46
2.45 - ../Rules/IRM_ID_00073.sch	47
2.46 - ../Rules/IRM_ID_00074.sch	48
2.47 - ../Rules/IRM_ID_00075.sch	49
2.48 - ../Rules/IRM_ID_00076.sch	50
2.49 - ../Rules/IRM_ID_00077.sch	51
2.50 - ../Rules/IRM_ID_00078.sch	52
2.51 - ../Rules/IRM_ID_00079.sch	54
2.52 - ../Rules/IRM_ID_00080.sch	55
2.53 - ../Rules/IRM_ID_00081.sch	56
2.54 - ../Rules/IRM_ID_00082.sch	57
2.55 - ../Rules/IRM_ID_00083.sch	58
2.56 - ../Rules/IRM_ID_00084.sch	59
2.57 - ../Rules/IRM_ID_00085.sch	60
2.58 - ../Rules/IRM_ID_00086.sch	61
2.59 - ../Rules/IRM_ID_00087.sch	62
2.60 - ../Rules/IRM_ID_00088.sch	63
2.61 - ../Rules/IRM_ID_00089.sch	64
2.62 - ../Rules/IRM_ID_00090.sch	65
2.63 - ../Rules/IRM_ID_00091.sch	66
2.64 - ../Rules/IRM_ID_00092.sch	67
2.65 - ../Rules/IRM_ID_00093.sch	68
2.66 - ../Rules/IRM_ID_00094.sch	69
2.67 - ../Rules/IRM_ID_00095.sch	70
2.68 - ../Rules/IRM_ID_00096.sch	71
2.69 - ../Rules/IRM_ID_00097.sch	72
2.70 - ../Rules/IRM_ID_00098.sch	73
2.71 - ../Rules/IRM_ID_00099.sch	74
2.72 - ../Rules/IRM_ID_00100.sch	75
2.73 - ../Rules/IRM_ID_00101.sch	76
2.74 - ../Rules/IRM_ID_00102.sch	77
2.75 - ../Rules/IRM_ID_00103.sch	78
2.76 - ../Rules/IRM_ID_00104.sch	79
2.77 - ../Rules/IRM_ID_00105.sch	80
Chapter 3 - Abstract Patterns	81
3.1 - ../Lib/CompareDateTimes.sch	82
3.2 - ../Lib/ICIdentifierRestrictions.sch	83
3.3 - ../Lib/IsmEnforcement.sch	84
3.4 - ../Lib/ValidateValueExistenceInList.sch	85
Chapter 4 - Min Accessible Rules	86
4.1 - ../Rules/IRM_ID_00062.sch	87
4.2 - ../Rules/IRM_ID_00063.sch	88
4.3 - ../Rules/IRM_ID_00064.sch	89
4.4 - ../Rules/IRM_ID_00065.sch	90
Chapter 5 - Schematron Schema	91
5.1 - ../IRM_XML.sch	92

Chapter 6 - Removed Rules	105
6.1 - ../Rules/deleted/IRM_ID_00001.sch	105
6.2 - ../Rules/deleted/IRM_ID_00003.sch	105
6.3 - ../Rules/deleted/IRM_ID_00004.sch	105
6.4 - ../Rules/deleted/IRM_ID_00008.sch	105
6.5 - ../Rules/deleted/IRM_ID_00009.sch	105
6.6 - ../Rules/deleted/IRM_ID_00011.sch	105
6.7 - ../Rules/deleted/IRM_ID_00012.sch	105
6.8 - ../Rules/deleted/IRM_ID_00013.sch	106
6.9 - ../Rules/deleted/IRM_ID_00014.sch	106
6.10 - ../Rules/deleted/IRM_ID_00018.sch	106
6.11 - ../Rules/deleted/IRM_ID_00026.sch	106
6.12 - ../Rules/deleted/IRM_ID_00027.sch	106
6.13 - ../Rules/deleted/IRM_ID_00028.sch	106
6.14 - ../Rules/deleted/IRM_ID_00031.sch	106
6.15 - ../Rules/deleted/IRM_ID_00032.sch	107
6.16 - ../Rules/deleted/IRM_ID_00035.sch	107
6.17 - ../Rules/deleted/IRM_ID_00037.sch	107
6.18 - ../Rules/deleted/IRM_ID_00038.sch	107
6.19 - ../Rules/deleted/IRM_ID_00039.sch	107
6.20 - ../Rules/deleted/IRM_ID_00049.sch	107
6.21 - ../Rules/deleted/IRM_ID_00056.sch	107
6.22 - ../Rules/deleted/IRM_ID_00057.sch	108
6.23 - ../Rules/deleted/IRM_ID_00058.sch	108
6.24 - ../Rules/deleted/IRM_ID_00060.sch	108
6.25 - ../Rules/deleted/IRM_ID_00061.sch	108
6.26 - ../Rules/deleted/IRM_ID_00066.sch	108
6.27 - ../Rules/deleted/IRM_ID_00067.sch	108
6.28 - ../Rules/deleted/IRM_ID_00069.sch	108

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for IRM. This guide is generated from the IRM 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 IRM. 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 IRM. The abstract patterns may be used in numbered rules or provided as reference for use in rules developed by users of IRM. 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 IRM are encoded using ISO Schematron. Schematron is a rule-based validation language that uses XML Path Language to make assertions about an XML document.

IRM 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 IRM's Schematron rules.

Chapter 2 - Rules

All of the numbered Rules for IRM 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 IRM-ID-XXXXX, with rule files named IRM_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/IRM_ID_00002.sch

Rule Description

[IRM-ID-00002][Error] For every attribute in the namespace [urn:us:gov:ic:irm] or [urn:us:mil:ces:metadata:ddms:5], a non-whitespace value must be specified. Human Readable:

Code Description

For each element which specifies an attribute in the IRM or DDMS namespace, we make sure that each attribute in the IRM or DDMS namespace specifies 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="IRM-ID-00002">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement[ddms:resource or irm:ICResourceMetadataPackage]//*[*[namespace-
uri()='urn:us:gov:ic:irm', 'urn:us:mil:ces:metadata:ddms:5']]">
    <sch:assert test="every $attribute in @[namespace-uri()='urn:us:gov:ic:irm', 'urn:us:mil:ces:metadata:ddms:5'] satisfies normalize-space(string($attribute))"
      flag="error">[IRM-ID-00002][Error] For every attribute in the namespace [urn:us:gov:ic:irm] or [urn:us:mil:ces:metadata:ddms:5], a non-whitespace value must
be specified. Human Readable:</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.2 - ../Rules/IRM_ID_00005.sch

Rule Description

[IRM-ID-00005][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639:digraph] then the value of attribute @ddms:value must be in CVENumIRMISO639Digraph.xml and no country code portion may be specified in the @ddms:language element value. Human Readable: ISO 639 digraph language codes must be in the ISO 639 digraph 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 (IRM_XML.sch), which reads values from a specific CVE file. This rule can directly check if the value of element Language is in the appropriate list because if a country code portion is specified in the element ddms:language's value, then the value of element ddms:language will not be found in the appropriate list and the assertion will fail as expected.

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="IRM-ID-00005" is-a="ValidateValueExistenceInList">
    <sch:param name="ruleText"
        value="" [IRM-ID-00005][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639:digraph] then the value of
attribute @ddms:value must be in CVENumIRMISO639Digraph.xml and no country code portion may be specified in the ddms:language element value. Human Readable: ISO 639 digraph language codes must
be in the ISO 639 digraph CVE. ""/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. ""/>
    <sch:param name="context"
        value="ddms:language[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:iso639:digraph']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$iso639DigraphList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00005][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639:digraph] then the value of
attribute @ddms:value must be in CVENumIRMISO639Digraph.xml and no country code portion may be specified in the ddms:language element value. Human Readable: ISO 639 digraph language codes must
be in the ISO 639 digraph CVE. ""/>
    </sch:pattern>
```

2.3 - ../Rules/IRM_ID_00006.sch

Rule Description

[IRM-ID-00006][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639-2:trigraph] then the value of attribute @ddms:value must be in CVENumIRMISO639-2Trigraph.xml and no country code portion may be specified in the ddms:value attribute value. Human Readable: ISO 639-2 trigraph language codes must be in the ISO 639-2 trigraph 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 (IRM_XML.sch), which reads values from a specific CVE file. This rule can directly check if the value of element ddms:language is in the appropriate list because if a country code portion is specified in the element ddms:language's value, then the value of element ddms:language will not be found in the appropriate list and the assertion will fail as expected.

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="IRM-ID-00006" is-a="ValidateValueExistenceInList">
    <sch:param name="ruleText"
        value="" [IRM-ID-00006][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639-2:trigraph] then the value
of attribute @ddms:value must be in CVENumIRMISO639-2Trigraph.xml and no country code portion may be specified in the ddms:value attribute value. Human Readable: ISO 639-2 trigraph language
codes must be in the ISO 639-2 trigraph CVE. '"/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="ddms:language[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:iso639-2:trigraph']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$iso639-2TrigraphList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00006][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639-2:trigraph] then the value
of attribute @ddms:value must be in CVENumIRMISO639-2Trigraph.xml and no country code portion may be specified in the ddms:value attribute value. Human Readable: ISO 639-2 trigraph language
codes must be in the ISO 639-2 trigraph CVE. '"/>
</sch:pattern>
```

2.4 - ../Rules/IRM_ID_00007.sch

Rule Description

[IRM-ID-00007][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639-3:trigraph] then the value of attribute @ddms:value must be in CVENumIRMISO639-3Trigraph.xml and no country code portion may be specified in the ddms:value attribute value. Human Readable: ISO 639-3 trigraph language codes must be in the ISO 639-3 trigraph 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 (IRM_XML.sch), which reads values from a specific CVE file. This rule can directly check if the value of element Language is in the appropriate list because if a country code portion is specified in the element ddms:language's value, then the value of element ddms:language will not be found in the appropriate list and the assertion will fail as expected.

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="IRM-ID-00007" is-a="ValidateValueExistenceInList">
    <sch:param name="ruleText"
        value="" [IRM-ID-00007][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639-3:trigraph] then the value
of attribute @ddms:value must be in CVENumIRMISO639-3Trigraph.xml and no country code portion may be specified in the ddms:value attribute value. Human Readable: ISO 639-3 trigraph language
codes must in the the ISO 639-3 trigraph CVE. '"/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="ddms:language[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:iso639-3:trigraph']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$iso639-3TrigraphList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00007][Error] If element ddms:language has the attribute @ddms:qualifier value of [urn:us:gov:ic:cvenum:irm:iso639-3:trigraph] then the value
of attribute @ddms:value must be in CVENumIRMISO639-3Trigraph.xml and no country code portion may be specified in the ddms:value attribute value. Human Readable: ISO 639-3 trigraph language
codes must in the the ISO 639-3 trigraph CVE. '"/>
</sch:pattern>
```

2.5 - ../Rules/IRM_ID_00010.sch

Rule Description

[IRM-ID-00010][Error] If element ddms:language has the attribute @ddms:qualifier value of [RFC5646] then the language code portion of the @ddms:value attribute value must be in CVENumIRMISO639Digraph.xml or CVENumIRMISO639-2Trigraph.xml and the country code portion, if present, must be in CVENumIRMCoverageISO3166Digraph.xml. Human Readable: RFC5646 language codes must comply with the RFC by using parts from ISO 639 Digraph or ISO 639-2 Trigraph and ISO 3166 Digraph.

Code Description

Finds ddms:language elements and checks its qualifier attribute for a value of [RFC5646]. If this value is found it will ensure that the value of the element's ddms:value attribute exists in the CVENumIRMISO639Digraph.xml or CVENumIRMISO639-2Trigraph.xml enumeration files represented by the \$iso639DigraphList or \$iso639-2TrigraphList variables. Country code portions (denoted by '-' separation) must be in the CVENumIRMCoverageISO3166Digraph.xml enumeration file represented by the \$coverageIso3166TrigraphList variable.

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="IRM-ID-00010">
    <sch:rule context="ddms:language[@ddms:qualifier='RFC5646']"><!-- Tokenize the element Language value into a list -->
        <sch:let name="tokens" value="tokenize(@ddms:value, '-')"/>
        <!-- For convenience and readability, save the primary and secondary subtags
              as defined in RFC 5646 -->
        <sch:let name="primarySubtag" value="$tokens[1]"/>
        <sch:let name="secondarySubtag" value="$tokens[2]"/>
        <sch:let name="badPrimaryValues"
            value="if($primarySubtag and ( index-of($iso639-2TrigraphList,lower-case($primarySubtag))>0 or index-of($iso639DigraphList,$primarySubtag)>0)) then null
else $primarySubtag"/>
        <sch:let name="badSecondaryValues"
            value="if($secondarySubtag and index-of($coverageIso3166DigraphList,$secondarySubtag)>0) then null else $secondarySubtag"/>
        <sch:let name="badValues"
            value="string-join(($badPrimaryValues, $badSecondaryValues), ' ')">
        <!-- Check if primary subtag is valid -->
        <sch:let name="primarySubtagValid"
            value="$primarySubtag and count($badPrimaryValues) = 0"/>
        <!-- Check if secondary subtag is valid -->
        <sch:let name="secondarySubtagValid"
            value="if(not($secondarySubtag)) then true() else count($badSecondaryValues) = 0"/>
        <sch:assert test="$primarySubtagValid and $secondarySubtagValid" flag="error">[IRM-ID-00010][Error] If element ddms:language has the attribute @ddms:qualifier value of
[RFC5646] then the language code portion of the @ddms:value attribute value must be in CVENumIRMISO639Digraph.xml or CVENumIRMISO639-2Trigraph.xml and the country code portion, if present,
must be in CVENumIRMCoverageISO3166Digraph.xml. Human Readable: RFC5646 language codes must comply with the RFC by using parts from ISO 639 Digraph or ISO 639-2 Trigraph and ISO 3166 Digraph.
The following values were found but are not in the CVEs:
        <sch:value-of select="for $each in tokenize($badValues, ' ') return concat(['',$each,'] ')">
        </sch:assert>
    </sch:rule>
</sch:pattern>
```

2.6 - ../Rules/IRM_ID_00015.sch

Rule Description

[IRM-ID-00015][Error] If element ddms:dates exists without one of the attributes @ddms:created or @ddms:posted Human Readable: Every ddms:dates element must have at least one of @ddms:created or @ddms:posted.

Code Description

This rule checks that for each occurrence of ddms:dates that either @ddms:created or @ddms:posted 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="IRM-ID-00015">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement/ddms:resource//ddms:dates">
        <sch:assert test="@ddms:created or @ddms:posted" flag="error">[IRM-ID-00015][Error] Every ddms:date must have at least one of @ddms:created or @ddms:posted.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.7 - ../Rules/IRM_ID_00016.sch

Rule Description

[IRM-ID-00016][Error] The permissible values for the year range are 1901 through the current year for attributes @ddms:approvedOn, @ddms:dateProcessed, @ddms:receivedOn, @ddms:infoCutOff, @ddms:posted, and @ddms:created. Human Readable: Dates must be after 1901 and in the past for @ddms:approvedOn, @ddms:dateProcessed, @ddms:receivedOn, @ddms:infoCutOff, @ddms:posted, and @ddms:created.

Code Description

This pattern uses abstract rules to consolidate logic. For attributes, we make sure 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="IRM-ID-00016"><!-- Use abstract rule to handle required attributes -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement/ddms:resource/*[@ddms:approvedOn | @ddms:dateProcessed | @ddms:receivedOn | @ddms:posted | @ddms:created | @ddms:infoCutOff]">
    <sch:let name="minYear" value="1901"/>
    <sch:let name="maxYear" value="$currentYear"/>
    <sch:let name="dateList"
        value="(string(@ddms:approvedOn), string(@ddms:dateProcessed), string(@ddms:receivedOn), string(@ddms:posted), string(@ddms:created),
string(@ddms:infoCutOff))"/>
    <sch:let name="errMsg"
        value="' [IRM-ID-00016][Error] The permissible values for the year range are 1901 through the current year for attributes @ddms:approvedOn,
@ddms:dateProcessed, @ddms:receivedOn, @ddms:infoCutOff, @ddms:posted, and @ddms:created. Human Readable: Dates must be after 1901 and in the past for @ddms:approvedOn, @ddms:dateProcessed,
@ddms:receivedOn, @ddms:infoCutOff, @ddms:posted, and @ddms:created. '"/>
        <sch:extends rule="abs.dateListYearRangeRule"/>
    </sch:rule>
</sch:pattern>
```

2.8 - ../Rules/IRM_ID_00017.sch

Rule Description

[IRM-ID-00017][Error] The permissible values for the year range are 1901 through 9999 for attribute @ddms:validTil. Human Readable: @ddms:validTil must be after 1901.

Code Description

This pattern uses abstract rules to consolidate logic. For attributes, we make sure 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="IRM-ID-00017"><!-- Use abstract rule to handle required attributes -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement/ddms:resource/*[@ddms:validTil]">
    <sch:let name="minYear" value="1901"/>
    <sch:let name="maxYear" value="9999"/>
    <sch:let name="dateList" value="(string(@ddms:validTil))"/>
    <sch:let name="errMsg"
        value="' [IRM-ID-00017][Error] The permissible values for the year range are 1901 through 9999 for attribute @ddms:validTil. Human Readable: @ddms:validTil
must be after 1901. '"/>
    <sch:extends rule="abs.dateListYearRangeRule"/>
</sch:rule>
</sch:pattern>
```

2.9 - ../Rules/IRM_ID_00019.sch

Rule Description

[IRM-ID-00019][Warning] @ddms:approvedOn must not be later than @ddms:created and @ddms:posted. Human Readable: The date held in the approvedOn attribute must not be later then the dates held by either the created or posted attributes.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param \$primaryDate 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 id="IRM-ID-00019" is-a="CompareDateTimes">
  <sch:param name="ruleText"
    value="" [IRM-ID-00019][Warning] @ddms:approvedOn must not be later than @ddms:created and @ddms:posted. ''/>
  <sch:param name="codeDesc"
    value="" This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $primaryDate 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="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement/ddms:resource//*[@ddms:approvedOn]"/>
  <sch:param name="primaryDate" value="@ddms:approvedOn"/>
  <sch:param name="operator" value="'<='"/>
  <sch:param name="secondaryDateList" value="(@ddms:created, @ddms:posted)">
  <sch:param name="flag" value="'error'"/>
</sch:pattern>
```

2.10 - ../Rules/IRM_ID_00020.sch

Rule Description

[IRM-ID-00020][Error] @ddms:infoCutOff must not be later than @ddms:created, and @ddms:posted. Human Readable: The date held in the infoCutOff attribute must not be later then the dates held by either the created or posted attributes.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param \$primaryDate 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 id="IRM-ID-00020" is-a="CompareDateTimes">
  <sch:param name="ruleText"
    value="' [IRM-ID-00020][Error] @ddms:infoCutOff must not be later than @ddms:created, and @ddms:posted. '"/>
  <sch:param name="codeDesc"
    value="' This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $primaryDate 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="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement/ddms:resource/*[@ddms:infoCutOff]"/>
  <sch:param name="primaryDate" value="@ddms:infoCutOff"/>
  <sch:param name="operator" value="'<='"/>
  <sch:param name="secondaryDateList" value="(@ddms:created, @ddms:posted)"/>
  <sch:param name="flag" value="'error'"/>
</sch:pattern>
```

2.11 - ../Rules/IRM_ID_00021.sch

Rule Description

[IRM-ID-00021][Warning] @ddms:validTil must not be earlier than @ddms:created, @ddms:posted, @ddms:infoCutOff, and @ddms:approvedOn. Human Readable: The date held by the validTil attribute must not be earlier than the dates in the created, posted, infoCutOff and approvedOn attributes.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param \$primaryDate 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 id="IRM-ID-00021" is-a="CompareDateTimes">
  <sch:param name="ruleText"
    value="" [IRM-ID-00021][Warning] @ddms:validTil must not be earlier than @ddms:created, @ddms:posted, @ddms:infoCutOff, and @ddms:approvedOn. '"/>
  <sch:param name="codeDesc"
    value="" This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $primaryDate 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="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement/ddms:resource/*[@ddms:validTil]"/>
  <sch:param name="primaryDate" value="@ddms:validTil"/>
  <sch:param name="operator" value="'>='"/>
  <sch:param name="secondaryDateList"
    value="(@ddms:created, @ddms:posted, @ddms:infoCutOff, @ddms:approvedOn)"/>
  <sch:param name="flag" value="'error'"/>
</sch:pattern>
```

2.12 - ../Rules/IRM_ID_00022.sch

Rule Description

[IRM-ID-00022][Error] For any element ddms:temporalCoverage, child element ddms:start must not be later than child element ddms:end. Human Readable: For date-time ranges, the start of a range must be earlier than, or equivalent to, the end of that range.

Code Description

This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param \$primaryDate 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 id="IRM-ID-00022" is-a="CompareDateTimes">
    <sch:param name="ruleText"
        value="" [IRM-ID-00022][Error] For any element ddms:temporalCoverage, child element ddms:start must not be later than child element ddms:end. ""/>
    <sch:param name="codeDesc"
        value="" This rule uses an abstract pattern to consolidate logic. It compares the date contained within the param $primaryDate 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="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage"/>
    <sch:param name="primaryDate" value="ddms:start"/>
    <sch:param name="operator" value="'<='"/>
    <sch:param name="secondaryDateList" value="(ddms:end)"/>
    <sch:param name="flag" value="'error'"/>
</sch:pattern>
```

2.13 - ../Rules/IRM_ID_00023.sch

Rule Description

[IRM-ID-00023][Error] The permissible values for the year range are 0001 through 9999 for elements ddms:start and ddms:end. Human Readable: ddms:start and ddms:end must be positive integers less than 10,000.

Code Description

This pattern uses abstract rules to consolidate logic. For attributes, we make sure 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="IRM-ID-00023"><!-- Use abstract rule to handle required attributes -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:start | tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/
tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:end">
    <sch:let name="minYear" value="0001"/>
    <sch:let name="maxYear" value="9999"/>
    <sch:let name="dateList" value="string(.)"/>
    <sch:let name="errMsg"
        value="' [IRM-ID-00023][Error] The permissible values for the year range are 0001 through 9999 for elements ddms:start and ddms:end. Human Readable: ddms:start
and ddms:end must be positive integers less than 10,000. '"/>
    <sch:extends rule="abs.dateListYearRangeRule"/>
</sch:rule>
</sch:pattern>
```

2.14 - ../Rules/IRM_ID_00024.sch

Rule Description

[IRM-ID-00024][Warning] For elements ddms:start and ddms:end and attributes @ddms:approvedOn, @ddms:dateProcessed, @ddms:receivedOn, @ddms:infoCutOff, @ddms:posted, @ddms:validTil, and @ddms:created, if the time designator (T) is specified, it is recommended that time zone be specified. Human Readable: For elements and attributes of date-time types, if the time designator (T) is specified, it is recommended that time zone be specified.

Code Description

The pattern applies to ddms:start and ddms:end elements, as well as any element that contains one or more attributes @ddms:approvedOn, @ddms:infoCutOff, @ddms:posted, and @ddms:created. It joins each of these attribute values, if present, into a larger space-separated string. It then breaks this string into tokens at each space character. If the value of the token contains the time zone designator (T), then it makes sure that the token value matches the regular expression for a timeZone, which is defined in the main schema file (IRM_XML.sch).

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="IRM-ID-00024"><!-- Abstract rule, which asserts that if the date $primaryDate specifies the
      time designator (T), then the timezone is specified -->
<sch:rule abstract="true" id="abs.rule00024">
    <sch:assert test="every $date in $dateList satisfies if($date castable as xs:dateTime and contains(string($date),'T')) then matches(string($date),
$endsWithTimeZoneRegEx) else true()"
                flag="warning">[IRM-ID-00024][Warning] For elements and attributes of date-time types, if the time designator (T) is specified, it is recommended that time
zone be specified.</sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule on required elements and attributes -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:start">
    <sch:let name="dateList" value="."/>
    <sch:extends rule="abs.rule00024"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:end">
    <sch:let name="dateList" value="."/>
    <sch:extends rule="abs.rule00024"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/* [@ddms:approvedOn | @ddms:dateProcessed |
@ddms:receivedOn | @ddms:posted | @ddms:created | @ddms:infoCutOff | @ddms:validTil]">
    <sch:let name="dateList"
            value="(@ddms:approvedOn, @ddms:dateProcessed, @ddms:receivedOn, @ddms:posted, @ddms:created, @ddms:infoCutOff, @ddms:validTil)"/>
    <sch:extends rule="abs.rule00024"/>
</sch:rule>
</sch:pattern>
```

2.15 - .//Rules/IRM_ID_00025.sch

Rule Description

[IRM-ID-00025][Error] The attribute @ism:excludeFromRollup must not be specified for any element in the namespace [urn:us:mil:ces:metadata:ddms:5] in a TDF IRM assertion. Human readable: DDMS 5.0 elements in IRM TDF assertions are not allowed to be excluded from roll-up because the security markings are now in the TDF assertion statement metadata.

Code Description

The attribute @ism:excludeFromRollup must not be specified for any element in the namespace [urn:us:mil:ces:metadata:ddms:5] in a TDF IRM 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="IRM-ID-00025">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement//ddms:*">
        <sch:assert test="not(@ism:excludeFromRollup)" flag="error">[IRM-ID-00025][Error]The attribute @ism:excludeFromRollup must not be specified for any element in the
namespace [urn:us:mil:ces:metadata:ddms:5] in a TDF IRM assertion. Human readable: DDMS 5.0 elements in IRM TDF assertions are not allowed to be excluded from roll-up because the security
markings are now in the TDF assertion statement metadata.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.16 - ../Rules/IRM_ID_00029.sch

Rule Description

[IRM-ID-00029][Error] If element ddms:geospatialCoverage has attribute @ddms:precedence with a value of [Secondary], there must be at least one sibling element ddms:geospatialCoverage for which attribute @ddms:precedence has a value of [Primary]. Human Readable: If a secondary country code is provided, there must also be a primary country code.

Code Description

If there is an element geospatialCoverage with attribute precedence specified with a value of [Secondary], then we make sure that there is at least one sibling geospatialCoverage element with attribute precedence specified with a value of [Primary].

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="IRM-ID-00029">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:geospatialCoverage[@ddms:precedence='Secondary']">
        <sch:assert test="../ddms:geospatialCoverage[@ddms:precedence='Primary']"
            flag="error">[IRM-ID-00029][Error] If element ddms:geospatialCoverage has attribute @ddms:precedence with a value of [Secondary], there must be at least one
sibling element ddms:geospatialCoverage for which attribute @ddms:precedence has a value of [Primary]. Human Readable: If a secondary country code is provided, there must also be a primary
country code.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.17 - ../Rules/IRM_ID_00030.sch

Rule Description

[IRM-ID-00030][Error] If attribute @ddms:order is specified with integer value N, there must exist other @ddms:order attributes with values 1 to N-1 with no duplicates. Human Readable: The values of attribute @ddms:order must be numbered sequentially with no duplicates, beginning at 1.

Code Description

A list, named \$orderList, is created containing the value of each order attribute within the document after normalizing to remove extra white-space. If the total number of items in \$orderList does not equal the number of distinct values in \$orderList, then a duplicate exists and we return false. Otherwise, we make sure that each number from 1 to N, where N is the number of items in \$orderList, is contained within \$orderList. If each number is contained, then we return true. Otherwise, false.

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="IRM-ID-00030">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource[//@ddms:order]">
        <sch:let name="orderList"
            value="tokenize(string-join(//@ddms:order/normalize-space(), ' '), ' ')" />
        <sch:assert test="(count(distinct-values($orderList)) = count($orderList) and (every $index in 1 to count($orderList) satisfies index-of($orderList,
xs:string($index))))"
            flag="error">[IRM-ID-00030][Error] If attribute @ddms:order is specified with integer value N, there must exist other @ddms:order attributes with values 1
to N-1 with no duplicates. Human Readable: The values of attribute @ddms:order must be numbered sequentially with no duplicates, beginning at 1.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.18 - ../Rules/IRM_ID_00033.sch

Rule Description

[IRM-ID-00033][Error] If element ddms:mimeType is specified, it must have a value from CVEnumMIMEType.xml. Human Readable: Values for ddms:mimeType must be defined in the MimeType 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 (IRM_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern id="IRM-ID-00033" is-a="ValidateValueExistenceInList">
  <sch:param name="ruleText"
    value="' [IRM-ID-00033][Error] If element ddms:mimeType is specified, it must have a value from CVEnumMIMEType.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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
  <sch:param name="context"
    value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:mimeType"/>
  <sch:param name="searchTerm" value="."/>
  <sch:param name="list" value="$mimeTypeList"/>
  <sch:param name="errMsg"
    value="' [IRM-ID-00033][Error] If element ddms:mimeType is specified, it must have a value from CVEnumMIMEType.xml. '"/>
</sch:pattern>
```

2.19 - ../Rules/IRM_ID_00034.sch

Rule Description

[IRM-ID-00034][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 (IRM_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern id="IRM-ID-00034" is-a="ValidateValueExistenceInList">
  <sch:param name="ruleText"
    value="" [IRM-ID-00034][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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
  <sch:param name="context"
    value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:language"/>
  <sch:param name="searchTerm" value="@ddms:qualifier"/>
  <sch:param name="list" value="$compoundLanguageQualifierTypeList"/>
  <sch:param name="errMsg"
    value="" [IRM-ID-00034][Error] For element ddms:language, attribute ddms:qualifier must have a value in CVEnumIRMCompoundLanguageQualifierType.xml. '"/>
</sch:pattern>
```

2.20 - ../Rules/IRM_ID_00036.sch

Rule Description

[IRM-ID-00036][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 XLink attributes are specified for an element, then the type and/or URL of the link must also 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="IRM-ID-00036">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]//tdf:StructuredStatement[ddms:resource|irm:ICResourceMetadataPackage]/*[@xlink:*)">
        <sch:assert test="normalize-space(string(@xlink:type)) or normalize-space(string(@xlink:href))"
            flag="error">[IRM-ID-00036][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 XLink attributes are specified for an element, then the type and/or URL of the link must also be specified.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.21 - ../Rules/IRM_ID_00040.sch

Rule Description

[IRM-ID-00040][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cvenum:irm:activity], then attribute ism:classification must also be specified. Human Readable: If the type element has the qualifier attribute with the value 'urn:us:gov:ic:cvenum:irm:activity' then the type element must also have the classification attribute that is not empty.

Code Description

For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cvenum:irm:activity], we make sure that attribute ism:classification 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 is-a="IsmEnforcement" id="IRM-ID-00040">
    <sch:param name="ruleText"
               value="[IRM-ID-00040][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cvenum:irm:activity], then attribute ism:classification must
also be specified."/>
    <sch:param name="codeDesc"
               value="' For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cvenum:irm:activity], we make sure that attribute
ism:classification is specified. '"/>
    <sch:param name="context"
               value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:activity']"/>
    <sch:param name="errMsg"
               value="' [IRM-ID-00040][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cvenum:irm:activity], then attribute ism:classification must
also be specified. '"/>
</sch:pattern>
```

2.22 - ../Rules/IRM_ID_00041.sch

Rule Description

IRM-ID-00041][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline], then attribute ism:classification must also be specified. Human Readable: If the type element has the qualifier attribute with the value 'urn:us:gov:ic:cenum:intdis:inteldiscipline' then the type element must also have the classification attribute that is not empty.

Code Description

For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cenum:intdis:inteldiscipline], we make sure that attribtue ism:classification 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 is-a="IsmEnforcement" id="IRM-ID-00041">
    <sch:param name="ruleText"
        value="[IRM-ID-00041][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline], then attribute
ism:classification must also be specified."/>
    <sch:param name="codeDesc"
        value="' For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cenum:intdis:inteldiscipline], we make sure that
attribtue ism:classification is specified.'"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cenum:intdis:inteldiscipline']"/>
    <sch:param name="errMsg"
        value="' [IRM-ID-00041][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline], then attribute
ism:classification must also be specified. '"/>
</sch:pattern>
```

2.23 - .//Rules/IRM_ID_00042.sch

Rule Description

[IRM-ID-00042][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component], then attribute ism:classification must also be specified. Human Readable: If the type element has the qualifier attribute with the value 'urn:us:gov:ic:cenum:intdis:inteldiscipline:component' then the type element must also have the classification attribute that is not empty.

Code Description

For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component], we make sure that attribtue ism:classification 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 is-a="IsmEnforcement" id="IRM-ID-00042">
    <sch:param name="ruleText"
        value="'[IRM-ID-00042][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component], then attribute
ism:classification must also be specified.'"/>
    <sch:param name="codeDesc"
        value="' For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component], we make
sure that attribtue ism:classification is specified.'"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cenum:intdis:inteldiscipline:component']"/>
    <sch:param name="errMsg"
        value="' [IRM-ID-00042][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component], then attribute
ism:classification must also be specified. '"/>
</sch:pattern>
```

2.24 - .//Rules/IRM_ID_00043.sch

Rule Description

If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique], then attribute ism:classification must also be specified. Human Readable: If the type element has the qualifier attribute with the value 'urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique' then the type element must also have the classification attribute that is not empty.

Code Description

For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique], we make sure that attribtue ism:classification 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 is-a="IsmEnforcement" id="IRM-ID-00043">
    <sch:param name="ruleText"
        value="'[IRM-ID-00043][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique], then
attribute ism:classification must also be specified.'"/>
    <sch:param name="codeDesc"
        value="' For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique],
we make sure that attribtue ism:classification is specified.'"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique']"/>
    <sch:param name="errMsg"
        value="' [IRM-ID-00043][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:cenum:intdis:inteldiscipline:component:technique], then
attribute ism:classification must also be specified. '"/>
</sch:pattern>
```

2.25 - ../Rules/IRM_ID_00044.sch

Rule Description

[IRM-ID-00044][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:irm:productline] then attribute ism:classification must also be specified. Human Readable: If the type element has the qualifier attribute with the value 'urn:us:gov:ic:irm:productline' then the type element must also have the classification attribute that is not empty.

Code Description

For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:irm:productline], we make sure that attribtue ism:classification 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 is-a="IsmEnforcement" id="IRM-ID-00044">
    <sch:param name="ruleText"
        value="[IRM-ID-00044][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:irm:productline] then attribute ism:classification must also
be specified."/>
    <sch:param name="codeDesc"
        value="' For each ddms:type element which specifies attribute ddms:qualifier with a value of [urn:us:gov:ic:irm:productline], we make sure that attribtue
ism:classification is specified.'"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:irm:productline']"/>
    <sch:param name="errMsg"
        value="' [IRM-ID-00044][Error] If element ddms:type is specified with a qualifier of [urn:us:gov:ic:irm:productline] then attribute ism:classification must also
be specified. '"/>
    </sch:pattern>
```

2.26 - ../Rules/IRM_ID_00045.sch

Rule Description

[IRM-ID-00045][Error] Element ddms:geospatialCoverage must have ISM classification markings. Human Readable: The geospatialCoverage element must have a classification attribute.

Code Description

For each ddms:geospatialCoverage element, we make sure that attribute ism:classification 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="IRM-ID-00045">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage">
        <sch:assert test="@ism:classification" flag="error">[IRM-ID-00045][Error] Element ddms:geospatialCoverage must have ISM classification markings. Human Readable: The
geospatialCoverage element must have a classification attribute.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.27 - ../Rules/IRM_ID_00046.sch

Rule Description

[IRM-ID-00046][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline:component:technique] the attribute @ddms:value must be in CVEnumIntelDisciplineComponentTechnique.xml. Human Readable: Intel Discipline Component Techniques must be in the CVEnumIntelDisciplineComponentTechnique 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 (IRM_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="IRM-ID-00046">
    <sch:param name="ruleText"
        value="" [IRM-ID-00046][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline:component:technique]
the attribute @ddms:value must be in CVEnumIntelDisciplineComponentTechnique.xml. Human Readable: Intel Discipline Component Techniques must be in the CVEnumIntelDisciplineComponentTechnique
CVE. ""/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. ""/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:intdis:inteldiscipline:component:technique']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$intelDisciplineComponentTechniqueList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00046][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline:component:technique]
the attribute @ddms:value must be in CVEnumIntelDisciplineComponentTechnique.xml. Human Readable: Intel Discipline Component Techniques must be in the CVEnumIntelDisciplineComponentTechnique
CVE. ""/>
    </sch:pattern>
```

2.28 - ../Rules/IRM_ID_00047.sch

Rule Description

[IRM-ID-00047][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline:component] the attribute @ddms:value must be in CVEnumIntelDisciplineComponent.xml. Human Readable: Intel Discipline Components must be in the CVEnumIntelDisciplineComponent 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 (IRM_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="IRM-ID-00047">
    <sch:param name="ruleText"
        value="" [IRM-ID-00047][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline:component] the
attribute @ddms:value must be in CVEnumIntelDisciplineComponent.xml. Human Readable: Intel Discipline Components must be in the CVEnumIntelDisciplineComponent CVE. '"/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:intdis:inteldiscipline:component']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$intelDisciplineComponentList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00047][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline:component] the
attribute @ddms:value must be in CVEnumIntelDisciplineComponent.xml. Human Readable: Intel Discipline Components must be in the CVEnumIntelDisciplineComponent CVE. '"/>
</sch:pattern>
```

2.29 - ./Rules/IRM_ID_00048.sch

Rule Description

[IRM-ID-00048][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline] the attribute @ddms:value must be in CVEnumIntelDiscipline.xml. Human Readable: Intel Disciplines must be in the CVEnumIntelDiscipline 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 (IRM_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="IRM-ID-00048">
    <sch:param name="ruleText"
        value="" [IRM-ID-00048][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline] the attribute
@ddms:value must be in CVEnumIntelDiscipline.xml. Human Readable: Intel Disciplines must be in the CVEnumIntelDiscipline CVE. '"/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:intdis:inteldiscipline']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$intelDisciplineList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00048][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:intdis:inteldiscipline] the attribute
@ddms:value must be in CVEnumIntelDiscipline.xml. Human Readable: Intel Disciplines must be in the CVEnumIntelDiscipline CVE. '"/>
</sch:pattern>
```

2.30 - .//Rules/IRM_ID_00050.sch

Rule Description

[IRM-ID-00050][Error] For element ddms:productionsMetrics, attribute @ddms:subject must be in CVEnumPMSubject.xml. Human Readable: Production Metric Subjects must be in the CVEnumPMSubject 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 (IRM_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="IRM-ID-00050">
    <sch:param name="ruleText"
        value="" [IRM-ID-00050][Error] For element ddms:productionsMetrics, attribute @ddms:subject must be in CVEnumPMSubject.xml. Human Readable: Production Metric
Subjects must be in the CVEnumPMSubject CVE. ""/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. ""/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:productionMetric[@ddms:subject]"/>
    <sch:param name="searchTerm" value="@ddms:subject"/>
    <sch:param name="list" value="$productionMetricsSubjectList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00050][Error] For element ddms:productionsMetrics, attribute @ddms:subject must be in CVEnumPMSubject.xml. Human Readable: Production Metric
Subjects must be in the CVEnumPMSubject CVE. ""/>
</sch:pattern>
```

2.31 - ../Rules/IRM_ID_00051.sch

Rule Description

[IRM-ID-00051][Error] For element ddms:productionsMetrics, attribute @ddms:coverage must be in CVEnumPMCoverage.xml. Human Readable: Production Metric Coverage values must be in the CVEnumPMCoverage 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 (IRM_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="IRM-ID-00051">
    <sch:param name="ruleText"
        value="" [IRM-ID-00051][Error] For element ddms:productionsMetrics, attribute @ddms:coverage must be in CVEnumPMCoverage.xml. Human Readable: Production Metric
Coverage values must be in the CVEnumPMCoverage CVE. ""/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. ""/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:productionMetric[@ddms:coverage]"/>
    <sch:param name="searchTerm" value="@ddms:coverage"/>
    <sch:param name="list" value="$productionMetricsCoverageList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00051][Error] For element ddms:productionsMetrics, attribute @ddms:coverage must be in CVEnumPMCoverage.xml. Human Readable: Production Metric
Coverage values must be in the CVEnumPMCoverage CVE. ""/>
</sch:pattern>
```

2.32 - .//Rules/IRM_ID_00052.sch

Rule Description

[IRM-ID-00052][Error] If element ddms:organization has attribute @ddms:acronym specified and the acronym does not contain a country component, then the value must be defined by the USAgency CES. Human Readable: Utilized agency acronyms with no country component must be defined in the USAgency CES.

Code Description

For ddms:organization with @ddms:acronym specified, if @ddms:acronym has no country component, then the country value must be defined by the USAgency CES.

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="IRM-ID-00052">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:organization[@ddms:acronym]">
        <sch:assert test="if (count(tokenize(normalize-space(@ddms:acronym),':')) = 1) then (some $token in $USAgencyAcronymList satisfies $token = normalize-
space(@ddms:acronym)) else true()"
                    flag="error">[IRM-ID-00052][Error] If element ddms:organization has attribute @ddms:acronym specified and the acronym does not contain a country component,
then the value must be defined by the USAgency CES. /&gt;</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.33 - ../Rules/IRM_ID_00053.sch

Rule Description

[IRM-ID-00053][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:activity] the attribute @ddms:value must be in CVEnumIRMActivity.xml. Human Readable: Activity must be in the CVEnumIRMActivity 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 (IRM_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern is-a="ValidateValueExistenceInList" id="IRM-ID-00053">
  <sch:param name="ruleText"
    value="" [IRM-ID-00053][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:activity] the attribute @ddms:value
must be in CVEnumIRMActivity.xml. Human Readable: Activity must be in the CVEnumIRMActivity CVE. '"/>
  <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
  <sch:param name="context"
    value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:activity']"/>
  <sch:param name="searchTerm" value="@ddms:value"/>
  <sch:param name="list" value="$activityList"/>
  <sch:param name="errMsg"
    value="" [IRM-ID-00053][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:activity] the attribute @ddms:value
must be in CVEnumIRMActivity.xml. Human Readable: Activity must be in the CVEnumIRMActivity CVE. '"/>
</sch:pattern>
```

2.34 - .//Rules/IRM_ID_00054.sch

Rule Description

[IRM-ID-00054][Error] If element ddms:geospatialCoverage has attribute @ddms:precedence specified, then the value must be in CVEnumIRMCoveragePrecedence.xml. Human Readable: Geospatial Coverage Precedence must be in the CVEnumIRMCoveragePrecedence 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 (IRM_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern is-a="ValidateValueExistenceInList" id="IRM-ID-00054">
  <sch:param name="ruleText"
    value="" [IRM-ID-00054][Error] If element ddms:geospatialCoverage has attribute @ddms:precedence specified, then the value must be in
CVEnumIRMCoveragePrecedence.xml. Human Readable: Geospatial Coverage Precedence must be in the CVEnumIRMCoveragePrecedence CVE. '"/>
  <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
  <sch:param name="context"
    value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage[@ddms:precedence]"/>
  <sch:param name="searchTerm" value="@ddms:precedence"/>
  <sch:param name="list" value="$coveragePrecedenceList"/>
  <sch:param name="errMsg"
    value="" [IRM-ID-00054][Error] If element ddms:geospatialCoverage has attribute @ddms:precedence specified, then the value must be in
CVEnumIRMCoveragePrecedence.xml. Human Readable: Geospatial Coverage Precedence must be in the CVEnumIRMCoveragePrecedence CVE. '"/>
</sch:pattern>
```

2.35 - .//Rules/IRM_ID_00055.sch

Rule Description

[IRM-ID-00055][Error] If ddms:geospatialCoverage/@order is specified then there must be one and only one of ddms:geospatialIdentifier/ddms:countryCode or ddms:geospatialIdentifier/ddms:subDivisionCode. Human Readable: A single order value must be applied to one country code or one subdivision code but not to both.

Code Description

Make sure that there is only one ddms:countryCode or order ddms:subDivisionCode when ddms:geospatialCoverage uses the order attribute.

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="IRM-ID-00055">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage[@ddms:order]">
        <sch:assert id="IRM-00055"
            test="count(ddms:geographicIdentifier/ddms:countryCode) + count(ddms:geographicIdentifier/ddms:subDivisionCode) = 1"
            flag="error">[IRM-ID-00055][Error] If ddms:geospatialCoverage/@order is specified then there must be one and only one of ddms:geospatialIdentifier/
ddms:countryCode or ddms:geospatialIdentifier/ddms:subDivisionCode. Human Readable: A single order value must be applied to one country code or one subdivision code</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.36 - ../Rules/IRM_ID_00059.sch

Rule Description

[IRM-ID-00059][Error] If MIN_DISCOVERABLE_OR_GREATER, element ddms:resource must specify at least one element ddms:subjectCoverage. Human Readable: If this IRM instance complies with the minimum discoverability rules, then the DDMS card must specify a subjectCoverage of the referenced resource.

Code Description

If MIN_DISCOVERABLE_OR_GREATER, we make sure that element ddms:resource specifies at least one element ddms:subjectCoverage.

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="IRM-ID-00059">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource[util:containsAnyOfTheTokens(@ddms:compliesWith,
('MIN_DISCOVERABLE'))]">
    <sch:assert test="ddms:subjectCoverage" flag="error">[IRM-ID-00059][Error] If MIN_DISCOVERABLE_OR_GREATER, element ddms:resource must specify at least one element
ddms:subjectCoverage. Human Readable: If this IRM instance complies with the minimum discoverability rules, then the DDMS card must specify a subjectCoverage of the referenced resource.</
sch:assert>
  </sch:rule>
</sch:pattern>
```

2.37 - ../Rules/IRM_ID_00062.sch

Rule Description

[IRM-ID-00062][Error] The value of an IC-ID identifier must follow standardized convention. Human Readable: The IC-ID identifier value has to follow standardized convention.

Code Description

This rule uses an abstract pattern that contains the logic for ensuring the value found if a given context exists, both provided as parameters from this implementation, follows the IC-ID identifier standardized convention.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00062" is-a="ICIdentifierRestrictions">
    <sch:param name="context"
               value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:identifier[@ddms:qualifier='IC-ID']"/>
    <sch:param name="value" value="string(@ddms:value)"/>
    <sch:param name="errorMessage"
               value="'[IRM-ID-00062][Error] The value of an IC-ID identifier must follow standardized convention. Human Readable: The IC-ID identifier value has to follow
standardized convention.'"/>
</sch:pattern>
```

2.38 - ../Rules/IRM_ID_00063.sch

Rule Description

[IRM-ID-00063][Error] Element ddms:resource/ddms:creator/ddms:organization must specify attribute @ddms:acronym. Human Readable: The DDMS card must specify a creator organization with an IC agency acronym for the referenced resource.

Code Description

We make sure that element ddms:resource/ddms:creator/ddms:organization exists and specifies attribute @ddms:acronym.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00063">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource">
        <sch:assert test="ddms:creator/ddms:organization/@ddms:acronym" flag="error">[IRM-ID-00063][Error] Element ddms:resource/ddms:creator/ddms:organization must specify
attribute @ddms:acronym. Human Readable: The DDMS card must specify a creator organization with an IC agency acronym for the referenced resource.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.39 - ../Rules/IRM_ID_00064.sch

Rule Description

[IRM-ID-00064][Error] Element ddms:resource/ddms:dates must specify attribute @ddms:created. Human Readable: The DDMS card must specify the date on which the referenced resource was created.

Code Description

We make sure that element ddms:resource/ddms:dates exists and specifies attribute @ddms:created.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00064">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource">
        <sch:assert test="ddms:dates/@ddms:created" flag="error">[IRM-ID-00064][Error] Element ddms:resource/ddms:dates must specify attribute @ddms:created. Human Readable:
The DDMS card must specify the date on which the referenced resource was created.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.40 - ../Rules/IRM_ID_00065.sch

Rule Description

[IRM-ID-00065][Error] Attribute ddms:resource/ddms:dates/@ddms:created must be castable as an xs:dateTime type. Human Readable: The date on which the referenced resource was created must be a dateTime type.

Code Description

We make sure that attribute dms:resource/ddms:dates/@ddms:created is castable as an xs:dateTime type.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00065">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:dates[@ddms:created]">
        <sch:assert test="@ddms:created castable as xs:dateTime" flag="error">[IRM-ID-00065][Error] Attribute ddms:resource/ddms:dates/@ddms:created must be castable as an
xs:dateTime type. Human Readable: The date on which the referenced resource was created must be a dateTime type.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.41 - ../Rules/IRM_ID_00068.sch

Rule Description

[IRM-ID-00068][Error] For element ddms:taskID, if attribute xlink:href exists, then the attribute must have a non-null value. Human Readable:

Code Description

The normalize-spaced value of attribute xlink:href is checked to make sure 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="IRM-ID-00068">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:taskID[@xlink:href]">
        <sch:assert test="normalize-space(string(@xlink:href))" flag="error">[IRM-ID-00068][Error] For element ddms:taskID if attribute xlink:href exists, then the attribute
must have a non-null value. Human Readable:</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.42 - ../Rules/IRM_ID_00070.sch

Rule Description

[IRM-ID-00070][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:executableindicator] the attribute @ddms:value must be in CVENumIRMExecutableIndicator.xml. Human Readable: Executable Indicator Value must be in the CVENumIRMExecutableIndicator 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 (IRM_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern is-a="ValidateValueExistenceInList" id="IRM-ID-00070">
  <sch:param name="ruleText"
    value="" [IRM-ID-00070][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:executableindicator] the attribute
@ddms:value must be in CVENumIRMExecutableIndicator.xml. Human Readable: Executable Indicator Value must be in the CVENumIRMExecutableIndicator CVE. '"/>
  <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
  <sch:param name="context"
    value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:executableindicator']"/>
  <sch:param name="searchTerm" value="@ddms:value"/>
  <sch:param name="list" value="$executableIndicatorList"/>
  <sch:param name="errMsg"
    value="" [IRM-ID-00070][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:executableindicator] the attribute
@ddms:value must be in CVENumIRMExecutableIndicator.xml. HHuman Readable: Executable Indicator Value must be in the CVENumIRMExecutableIndicator CVE. '"/>
</sch:pattern>
```

2.43 - ../Rules/IRM_ID_00071.sch

Rule Description

[IRM-ID-00071][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:maliciouscodeindicator] the attribute @ddms:value must be in CVENumIRMMaliciousCodeIndicator.xml. Human Readable: Malicious Code Indicator values must be in the CVENumIRMMaliciousCodeIndicator 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 (IRM_XML.sch), which reads values from a specific CVE file.

Schematron Code

```
<sch:pattern is-a="ValidateValueExistenceInList" id="IRM-ID-00071">
  <sch:param name="ruleText"
    value="" [IRM-ID-00071][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:maliciouscodeindicator] the attribute
@ddms:value must be in CVENumIRMMaliciousCodeIndicator.xml. Human Readable: Malicious Code Indicator values must be in the CVENumIRMMaliciousCodeIndicator CVE. ''/>
  <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. ''/>
  <sch:param name="context"
    value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:maliciouscodeindicator']"/>
  <sch:param name="searchTerm" value="@ddms:value"/>
  <sch:param name="list" value="$maliciousCodeIndicatorList"/>
  <sch:param name="errMsg"
    value="" [IRM-ID-00071][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:maliciouscodeindicator] the attribute
@ddms:value must be in CVENumIRMMaliciousCodeIndicator.xml. Human Readable: Malicious Code Indicator values must be in the CVENumIRMMaliciousCodeIndicator CVE. ''/>
</sch:pattern>
```

2.44 - ../Rules/IRM_ID_00072.sch

Rule Description

[IRM-ID-00072][Error] For element ddms:searchableDate, ddms:start must be earlier than ddms:end. Human Readable: Within the searchableDate element, the date within the start element must be earlier than the date within the end element.

Code Description

For each element ddms:searchableDate, the child elements ddms:start and ddms:end are cast as xs:dateTime types, then compared to ensure start is less than end.

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="IRM-ID-00072">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:searchableDate">
        <sch:assert test="if((ddms:start castable as xs:dateTime) and (ddms:end castable as xs:dateTime)) then xs:dateTime(ddms:start) < xs:dateTime(ddms:end) else false()"
            flag="error">[IRM-ID-00072][Error] For element ddms:searchableDate, ddms:start must be earlier than ddms:end. Human Readable: Within the searchableDate
element, the date within the start element must be earlier than the date within the end element.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.45 - ../Rules/IRM_ID_00073.sch

Rule Description

[IRM-ID-00073][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:positive:intel] the attribute @ddms:value must be in CVENumIRMPositiveIntel.xml. Human Readable: Positive Intel values must be in the CVENumIRMPositiveIntel 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 (IRM_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="IRM-ID-00073">
    <sch:param name="ruleText"
        value="" [IRM-ID-00073][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:positive:intel] the attribute
@ddms:value must be in CVENumIRMPositiveIntel.xml. Human Readable: Positive Intel values must be in the CVENumIRMPositiveIntel CVE. '"/>
    <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 $list is a variable defined in the main document (IRM_XML.sch), which reads values from a
specific CVE file. '"/>
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:type[@ddms:qualifier='urn:us:gov:ic:cvenum:irm:positive:intel']"/>
    <sch:param name="searchTerm" value="@ddms:value"/>
    <sch:param name="list" value="$positiveIntelList"/>
    <sch:param name="errMsg"
        value="" [IRM-ID-00073][Error] If element ddms:type has attribute @ddms:qualifier specified as [urn:us:gov:ic:cvenum:irm:positive:intel] the attribute
@ddms:value must be in CVENumIRMPositiveIntel.xml. Human Readable: Positive Intel values must be in the CVENumIRMPositiveIntel CVE. '"/>
</sch:pattern>
```

2.46 - ../Rules/IRM_ID_00074.sch

Rule Description

[IRM-ID-00074][Error] For element ddms:searchableDate, elements ddms:start and ddms:end must match the xsd:dateTime format. Human Readable: Within the searchableDate element, the start and end elements values must conform to the xsd:dateTime format.

Code Description

For each element ddms:searchableDate, we make sure that elements ddms:start and ddms:end are each castable as xs:dateTime type.

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="IRM-ID-00074">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:searchableDate">
        <sch:assert test="(ddms:start castable as xs:dateTime) and (ddms:end castable as xs:dateTime)"
            flag="error">[IRM-ID-00074][Error] For element ddms:searchableDate, elements ddms:start and ddms:end must match the xsd:dateTime format. Human Readable:
Within the searchableDate element, the start and end elements values must conform to the xsd:dateTime format.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.47 - ../Rules/IRM_ID_00075.sch

Rule Description

[IRM-ID-00075][Error] For element ddms:temporalCoverage, at least one of its child elements, other than ddms:name, must be specified. Human Readable: Within the temporalCoverage element, at least one of its child elements must not be blank with the exception of the name element.

Code Description

This rule ensures that for each ddms:temporalCoverage element one or more child elements contains non-whitespace content by making sure the string representation of ddms:temporalCoverage, which includes all children except ddms:name, contains 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="IRM-ID-00075">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement//ddms:resource//ddms:temporalCoverage">
        <sch:assert test="count(descendant::node())[not(self::ddms:name)][normalize-space(string-join(text(),''))] > 0"
            flag="error">[IRM-ID-00075][Error] For element ddms:temporalCoverage, at least one of its child elements, other than ddms:name, must be specified. Human
Readable: Within the temporalCoverage element, at least one of it's child elements must not be blank with the exception of the name element.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.48 - ../Rules/IRM_ID_00076.sch

Rule Description

[IRM-ID-00076][Error] If the ddms:acquiredOn element exists, at least one of its child elements ddms:description, ddms:approximableDate, or ddms:searchableDate must be present. Human Readable: The acquiredOn element must have at least one of the following child elements: description, approximableDate and searchableDate.

Code Description

For element ddms:acquiredOn, we make sure that one or more of the child elements ddms:description, ddms:approximableDate, ddms:searchableDate/ddms:start, or ddms:searchableDate/ddms:end is specified with 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="IRM-ID-00076">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:acquiredOn">
        <sch:assert test="normalize-space(string(ddms:description)) or normalize-space(string(ddms:approximableDate)) or normalize-space(string(ddms:searchableDate/ddms:start))
or normalize-space(string(ddms:searchableDate/ddms:end))"
                    flag="error">[IRM-ID-00076][Error] If the ddms:acquiredOn element exists, at least one of its child elements ddms:description, ddms:approximableDate, or
ddms:searchableDate must be present. Human Readable: The acquiredOn element must have at least one of the following child elements: description, approximableDate and searchableDate.</
sch:assert>
    </sch:rule>
</sch:pattern>
```

2.49 - ../Rules/IRM_ID_00077.sch

Rule Description

[IRM-ID-00077][Error] For element ddms:person at least one of the following child elements must have non-whitespace content: ddms:surname, ddms:userID, ddms:name, ddms:affiliation, ddms:postalAddress, ddms:phone ddms:email. Human Readable:

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 ddms:postalAddress cannot contain text content, so we count the number of its child elements that contain non-white space and make sure that the count is great than 0. The abstract rule is extended once for each required element in rule IRM_ID_00077.

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="IRM-ID-00077"><!-- Abstract rule, which asserts that at least one of the listed child elements has non-whitespace content -->
<sch:rule abstract="true" id="abs.rule00077">
    <sch:assert test="ddms:surname[normalize-space(string(text()))] or ddms:userID[normalize-space(string(text()))] or ddms:name[normalize-space(string(text()))] or
ddms:affiliation[normalize-space(string(text()))] or ddms:phone[normalize-space(string(text()))] or ddms:email[normalize-space(string(text()))] or (some $token in
ancestor::ddms:postalAddress/*/text() satisfies normalize-space(string($token)))"
        flag="error">[IRM-ID-00077][Error] For element ddms:person at least one of the following child elements must have non-whitespace content: ddms:surname,
ddms:userID, ddms:name, ddms:affiliation, ddms:postalAddress, ddms:phone ddms:email. Human Readable:</sch:assert>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:person">
    <sch:extends rule="abs.rule00077"/>
    </sch:rule>
</sch:pattern>
```

2.50 - .//Rules/IRM_ID_00078.sch

Rule Description

[IRM-ID-00078][Error] For elements ddms:acquiredOn and ddms:temporalCoverage with child element name [infoCutoff], the permissible values for the year range are 1901 through the current year for elements ddms:approximableDate, ddms:start, and ddms:end. Human Readable: If elements acquiredOn and temporalCoverage have a child element infoCutoff, then the approximableDate, start and end elements must have a year value between 1901 and the current year.

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 IRM-ID-00078.

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="IRM-ID-00078">
    <sch:rule abstract="true" id="abs.rule00078">
        <sch:let name="minYear" value="1901"/>
        <sch:let name="maxYear" value="$currentYear"/>
        <sch:let name="dateValue" value="."/>
        <sch:let name="errMsg"
            value="" [PUBS-ID-00078][Error] For elements ddms:acquiredOn and ddms:temporalCoverage with child element name [infoCutoff], the permissible values for the
year range are 1901 through the current year for elements ddms:approximableDate, ddms:start, and ddms:end. ''/>
        <sch:extends rule="abs.dateYearRangeRule"/>
    </sch:rule>
    <!-- Begin using abstract rule to check required elements -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage[ddms:name='infoCutoff']/ddms:approximableStart/
ddms:searchableDate/ddms:start">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage[ddms:name='infoCutoff']/
ddms:approximableStart/ddms:searchableDate/ddms:end">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage[ddms:name='infoCutoff']/
ddms:approximableEnd/ddms:searchableDate/ddms:start">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage[ddms:name='infoCutoff']/
ddms:approximableEnd/ddms:searchableDate/ddms:end">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage[ddms:name='infoCutoff']/
ddms:start">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:temporalCoverage[ddms:name='infoCutoff']/
ddms:end">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:acquiredOn/ddms:approximableDate">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:acquiredOn/ddms:searchableDate/ddms:start">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:acquiredOn/ddms:searchableDate/ddms:end">
    <sch:extends rule="abs.rule00078"/>
</sch:rule>
</sch:pattern>
```

2.51 - ../Rules/IRM_ID_00079.sch

Rule Description

[IRM-ID-00079][Error] The @icid:DESVersion is less than the minimum version allowed: 1. Human Readable: The IC-ID version imported by IRM must be greater than or equal to 1.

Code Description

For all irm:ICResourceMetadataPackage elements that contain @icid:DESVersion, we verify 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="IRM-ID-00079">
    <sch:rule context="irm:ICResourceMetadataPackage[@icid:DESVersion]">
        <sch:let name="version"
            value="number(if (contains(@icid:DESVersion,'-')) then substring-before(@icid:DESVersion,'-') else @icid:DESVersion)"/>
        <sch:assert test="$version >= 1" flag="error">[IRM-ID-00079][Error] The @icid:DESVersion is less than the minimum version allowed: 1. Human Readable: The IC-ID
version imported by IRM must be greater than or equal to 1.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.52 - ../Rules/IRM_ID_00080.sch

Rule Description

[IRM-ID-00080][Error] The @usagency:CESVersion is less than the minimum version allowed: 201502. Human Readable: The US Agency version imported by IRM must be greater than or equal to 2015-FEB.

Code Description

For all irm:ICResourceMetadataPackage elements that contain @usagency:CESVersion, we verify that the version is greater than or equal to the minimum allowed version: 201502.

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="IRM-ID-00080">
    <sch:rule context="irm:ICResourceMetadataPackage[@usagency:CESVersion]">
        <sch:let name="version"
            value="number(if (contains(@usagency:CESVersion,'-')) then substring-before(@usagency:CESVersion,'-') else @usagency:CESVersion)"/>
        <sch:assert test="$version >= 201502" flag="error">[IRM-ID-00080][Error] The @usagency:CESVersion is less than the minimum version allowed: 201502. Human Readable:
The US Agency version imported by IRM must be greater than or equal to 2015-FEB.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.53 - ../Rules/IRM_ID_00081.sch

Rule Description

[IRM-ID-00081][Error] A ddms:type element with @ddms:qualifer ProductLine or Intel must not contain any text. Human Readable: DDMS Types of ProductLine or Intel must not contain any text within the element.

Code Description

For all ddms:type elements that contain a @ddms:qualifier of urn:us:gov:ic:irm:productline or that start with urn:us:gov:ic:cvenum:intdis, we verify that the element does not contain any text.

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="IRM-ID-00081">
    <sch:rule context="ddms:type[@ddms:qualifier='urn:us:gov:ic:irm:productline'] | ddms:type[contains(@ddms:qualifier,'urn:us:gov:ic:cvenum:intdis')]">
        <sch:assert test="not(normalize-space(text()))" flag="error">[IRM-ID-00081][Error] A ddms:type element with @ddms:qualifer ProductLine or Intel must not contain any
text. Human Readable: DDMS Types of ProductLine or Intel must not contain any text within the element.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.54 - ../Rules/IRM_ID_00082.sch

Rule Description

[IRM-ID-00082][Error] Use of ddms:revisionRecall is forbidden in IRM and a RevRecall assertion should be used instead. Human Readable: Use of the revisionRecall element in DDMS is forbidden with the use of IRM and a Revision/Recall assertion should be used instead.

Code Description

If there is an IRM assertion, and a peer DDMS assertion has a ddms:revisionRecall element then trigger the 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="IRM-ID-00082"><!-- Abstract rule, which asserts that at least one of the listed child elements has non-whitespace content -->
<sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:revisionRecall">
    <sch:assert test="false()" flag="error">[IRM-ID-00082][Error] Use of ddms:revisionRecall is forbidden in IRM and a RevRecall assertion should be used instead. Human
Readable: Use of the revisionRecall element in DDMS is forbidden with the use of IRM and a Revision/Recall assertion should be used instead.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.55 - ../Rules/IRM_ID_00083.sch

Rule Description

[IRM-ID-00083][Error] If an IRM TrustedDataObject structured statement exists, it must have a DDMS assertion. Human Readable: If an IRM Trusted Data Object structured statement exists, it must have a DDMS assertion.

Code Description

If a TDO has an IRM structured statement, then verify that it has a DDMS 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="IRM-ID-00083">
    <sch:rule context="tdf:TrustedDataObject/tdf:Assertion/tdf:StructuredStatement/*[namespace-uri()='urn:us:gov:ic:irm' and local-name()='ICResourceMetadataPackage']">
        <sch:assert test="ancestor::tdf:TrustedDataObject/tdf:Assertion/tdf:StructuredStatement/*[namespace-uri()='urn:us:mil:ces:metadata:ddms:5' and local-name()='resource']"
            flag="error">[IRM-ID-00083][Error] If a IRM tdf:TrustedDataObject structured statement exists, then it must have a DDMS assertion. Human Readable: If a
Trusted Data Object has an IRM structured statement, it must have a DDMS assertion.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.56 - ../Rules/IRM_ID_00084.sch

Rule Description

[IRM-ID-00084][Error] If @ddms:subject or @ddms:coverage exists, then @pm:CESVersion must exist as well.

Code Description

Make sure that the PM CVE version attribute exists if subject and coverage 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="IRM-ID-00084">
    <sch:rule context="*[@ddms:subject|@ddms:coverage]">
        <sch:assert test="//@pm:CESVersion" flag="error">[IRM-ID-00084][Error] If @ddms:subject or @ddms:coverage exists, then @pm:CESVersion must exist as well.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.57 - ../Rules/IRM_ID_00085.sch

Rule Description

[IRM-ID-00085][Error] The @pm:CESVersion is less than the minimum version allowed: 201511. Human Readable: The PM version imported by IRM must be greater than or equal to 2015-NOV.

Code Description

For all elements that contain @pm:CESVersion, this rule verifies that the version is greater than or equal to the minimum allowed version: 201511.

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="IRM-ID-00085">
    <sch:rule context="*[@pm:CESVersion]">
        <sch:let name="version"
            value="number(if (contains(@pm:CESVersion,'-')) then substring-before(@pm:CESVersion,'-') else @pm:CESVersion)"/>
        <sch:assert test="$version >= 201511" flag="error">[IRM-ID-00085][Error] The @pm:CESVersion is less than the minimum version allowed: 201511. Human Readable: The PM
version imported by IRM must be greater than or equal to 2015-NOV.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.58 - ../Rules/IRM_ID_00086.sch

Rule Description

[IRM-ID-00086][Error] The @intdis:CESVersion is less than the minimum version allowed: 201609. Human Readable: The INTDIS version imported by IRM must be greater than or equal to 2016-SEP.

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: 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="IRM-ID-00086">
    <sch:rule context="irm:ICResourceMetadataPackage[@intdis:CESVersion]">
        <sch:let name="version"
            value="number(if (contains(@intdis:CESVersion,'-')) then substring-before(@intdis:CESVersion,'-') else @intdis:CESVersion)"/>
        <sch:assert test="$version >= 201609" flag="error">[IRM-ID-00086][Error] The @intdis:CESVersion is less than the minimum version allowed: 201609. Human Readable: The
INTDIS version imported by IRM must be greater than or equal to 2016-SEP.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.59 - ../Rules/IRM_ID_00087.sch

Rule Description

[IRM-ID-00087][Error] If @ddms:qualifier identifies a intelligence discipline URN, then @intdis:CESVersion must exist as well.

Code Description

Make sure that the INTDIS CVE version attribute exists if intelligence discipline resources are identified.

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="IRM-ID-00087">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:type[starts-
with(@ddms:qualifier,'urn:us:gov:ic:cvenum:intdis:intelldiscipline')]">
    <sch:assert test="ancestor-or-self::tdf:*/@intdis:CESVersion" flag="error">[IRM-ID-00087][Error] If @ddms:qualifier identifies a intelligence discipline URN, then
@intdis:CESVersion must exist as well.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.60 - ../Rules/IRM_ID_00088.sch

Rule Description

[IRM-ID-00088][Error] The @mime:CESVersion is less than the minimum version allowed: 201609. Human Readable: The MIME version imported by IRM 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="IRM-ID-00088">
  <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">[IRM-ID-00088][Error] The @mime:CESVersion is less than the minimum version allowed: 201609. Human Readable: The
MIME version imported by IRM must be greater than or equal to 2016-SEP.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.61 - ../Rules/IRM_ID_00089.sch

Rule Description

[IRM-ID-00089][Error] If @ddms:mimeType exists, then @mime:CESVersion must exist as well.

Code Description

Make sure that the MIME CVE version attribute exists if the internet media type of the resource is defined.

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="IRM-ID-00089">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:mimeType">
        <sch:assert test="//@mime:CESVersion" flag="error">[IRM-ID-00089][Error] If @ddms:mimeType exists, then @mime:CESVersion must exist as well.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.62 - ../Rules/IRM_ID_00090.sch

Rule Description

[IRM-ID-00090][Error] If subDivisionCode codespace is GENC codespace, then value must be in the GENC subDivisionCode cve.

Code Description

If subDivisionCode codespace is GENC codespace, then value must be in the GENC subDivisionCode cve.

Schematron Code

```
<sch:pattern id="IRM-ID-00090">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage//ddms:subDivisionCode">
    <sch:let name="isGENCSubDivision"
      value="matches(normalize-space(./@ddms:codespace), '^as:GENC:6:(ed3|3-[1-9][0-9]*)$')"/>
    <sch:assert test="not($isGENCSubDivision) or (some $token in $gencSubDivisionList satisfies $token = normalize-space(./@ddms:code))"
      flag="error">[IRM-ID-00090][Error] If subDivisionCode codespace is GENC codespace, then value must be in the GENC subDivisionCode cve.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.63 - ../Rules/IRM_ID_00091.sch

Rule Description

[IRM-ID-00091][Warning] Deprecated MIME types should not be used.

Code Description

Deprecated MIME types should not be used.

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="IRM-ID-00091">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:mimeType">
    <sch:assert test="not(some $deprecatedMimeTypeValue in $deprecatedMimeTypeList satisfies . = $deprecatedMimeTypeValue)"
              flag="error">[IRM-ID-00091][Warning] Deprecated MIME types should not be used.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.64 - ../Rules/IRM_ID_00092.sch

Rule Description

[IRM-ID-00092][Error] ddms:NonStateActor should contain a value from the NonStateActor CVE

Code Description

Make sure that the value within NonStateActor is a value from the CVE.

Schematron Code

```
<sch:pattern id="IRM-ID-00092">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//
ddms:nonStateActor[@ddms:qualifier='urn:us:gov:ic:cvenum:pm:nonstateactors']">
    <sch:assert test="some $token in $nonStateActorsList satisfies $token = normalize-space(./text())"
              flag="error">[IRM-ID-00092][Error] ddms:NonStateActor should contain a value from the NonStateActor CVE</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.65 - ../Rules/IRM_ID_00093.sch

Rule Description

[IRM-ID-00093][Error] If present, at least one ddms:countryCode in a ddms:geographicIdentifier must be a GENC ED3 codespace: ^ge:GENC:3:(ed3|3-[1-9][0-9]*)\$

Code Description

If present, at least one ddms:countryCode in a ddms:geographicIdentifier must be a GENC ED3 codespace: ^ge:GENC:3:(ed3|3-[1-9][0-9]*)\$

Schematron Code

```
<sch:pattern id="IRM-ID-00093">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage">
    <sch:let name="hasCountryCodes"
      value="count(ddms:geographicIdentifier/ddms:countryCode) > 0"/>
    <sch:assert test="not($hasCountryCodes) or (some $countryCode in ddms:geographicIdentifier/ddms:countryCode satisfies matches(normalize-space($countryCode/
@ddms:codespace), '^ge:GENC:3:(ed3|3-[1-9][0-9]*)$'))"
      flag="error">[IRM-ID-00093][Error] ddms:countryCode must be a GENC ED3 codespace: ^ge:GENC:3:(ed3|3-[1-9][0-9]*)$</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.66 - ../Rules/IRM_ID_00094.sch

Rule Description

[IRM-ID-00094][Error] If countrycode codespace is GENC codespace, then value must be in the GENC countrycode cve.

Code Description

If countrycode codespace is GENC codespace, then value must be in the GENC countrycode cve

Schematron Code

```
<sch:pattern id="IRM-ID-00094">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage//ddms:countryCode">
    <sch:let name="isGENCCountryCode"
      value="matches(normalize-space(./@ddms:codespace), '^ge:GENC:3:(ed3|3-[1-9][0-9]*)$')"/>
    <sch:assert test="not($isGENCCountryCode) or (some $token in $gencCountryCodeList satisfies $token = normalize-space(./@ddms:code))"
      flag="error">[IRM-ID-00094][Error] ddms:countryCode must be in GENC countrycode cve.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.67 - ../Rules/IRM_ID_00095.sch

Rule Description

[IRM-ID-00095][Error] If present, at least one ddms:subDivisionCode in a ddms:geographicIdentifier must be a GENC ED3 codespace: ^as:GENC:6:(ed3|3-[1-9][0-9]*)\$

Code Description

If present, at least one ddms:subDivisionCode in a ddms:geographicIdentifier must be a GENC ED3 codespace: ^as:GENC:6:(ed3|3-[1-9][0-9]*)\$

Schematron Code

```
<sch:pattern id="IRM-ID-00095">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:geospatialCoverage">
    <sch:let name="hasSubDivisions"
      value="count(ddms:geographicIdentifier/ddms:subDivisionCode) > 0"/>
    <sch:assert test="not($hasSubDivisions) or (some $subDivisionCode in ddms:geographicIdentifier/ddms:subDivisionCode satisfies matches(normalize-space($subDivisionCode/
@ddms:codespace), '^as:GENC:6:(ed3|3-[1-9][0-9]*)$'))"
      flag="error">[IRM-ID-00095][Error] ddms:subDivisionCode must be a GENC ED3 codespace: ^as:GENC:6:(ed3|3-[1-9][0-9]*)$</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.68 - ../Rules/IRM_ID_00096.sch

Rule Description

[IRM-ID-00096][Error] The @ism:DESVersion is less than the minimum version allowed: 13. Human Readable: The ISM version imported by IRM 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="IRM-ID-00096">
  <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">[IRM-ID-00096][Error] The @ism:DESVersion is less than the minimum version allowed: 13. Human Readable: The ISM
version imported by IRM must be greater than or equal to 13.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.69 - ../Rules/IRM_ID_00097.sch

Rule Description

[IRM-ID-00097][Error] The @tdf:version is less than the minimum version allowed: 3. Human Readable: The TDF version imported by IRM 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="IRM-ID-00097">
  <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">[IRM-ID-00097][Error] The @tdf:version is less than the minimum version allowed: 3. Human Readable: The TDF version
imported by IRM must be greater than or equal to 3.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.70 - ../Rules/IRM_ID_00098.sch

Rule Description

[IRM-ID-00098][Error] Use of a GENC codespace requires the presence of the IC-GENC CESVersion attribute.

Code Description

If a codespace attribute is specified that contains a GENC codespace, then ensure that the IC-GENC CESVersion attribute is specified in the IRM assertion on the ICResourceMetadataPackage.

Schematron Code

```
<sch:pattern id="IRM-ID-00098">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//*[starts-with(@ddms:codespace,'ge:GENC') or
starts-with(@ddms:codespace,'as:GENC')]">
    <sch:assert test="ancestor-or-self::tdf:*/tdf:Assertion//irm:ICResourceMetadataPackage/@genc:CESVersion"
              flag="error">[IRM-ID-00098][Error] Use of a GENC codespace requires the presence of the IC-GENC CESVersion attribute.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.71 - ../Rules/IRM_ID_00099.sch

Rule Description

[IRM-ID-00099][Error] The @genc:CESVersion is less than the minimum version allowed: 201609. Human Readable: The IC-GENC version imported by IRM must be greater than or equal to 2016-SEP.

Code Description

For all irm:ICResourceMetadataPackage elements that contain @genc:CESVersion, we verify 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="IRM-ID-00099">
    <sch:rule context="irm:ICResourceMetadataPackage[@genc:CESVersion]">
        <sch:let name="version"
            value="number(if (contains(@genc:CESVersion,'-')) then substring-before(@genc:CESVersion,'-') else @genc:CESVersion)"/>
        <sch:assert test="$version >= 201609" flag="error">[IRM-ID-00099][Error] The @genc:CESVersion is less than the minimum version allowed: 201609. Human Readable: The
IC-GENC version imported by IRM must be greater than or equal to 2016-SEP.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.72 - ../Rules/IRM_ID_00100.sch

Rule Description

[IRM-ID-00100][Error] If a ddms:resource element is present in a tdf:StructuredStatement within a tdf:Assertion, then the tdf:Assertion must also contain a tdf:StatementMetadata element.

Code Description

If a ddms:resource element is present in a tdf:StructuredStatement within a tdf:Assertion, then the tdf:Assertion must also contain a tdf:StatementMetadata element.

Schematron Code

```
<sch:pattern id="IRM-ID-00100">
  <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource">
    <sch:let name="hasStatementMetadata"
      value="count(..preceding-sibling::tdf:StatementMetadata) > 0"/>
    <sch:assert test="$hasStatementMetadata" flag="error">[IRM-ID-00100][Error] tdf:Assertion must contain tdf:StatementMetadata when tdf:StructuredStatement contains
ddms:resource.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.73 - .//Rules/IRM_ID_00101.sch

Rule Description

[IRM-ID-00101][Error] If element ddms:organization has attribute @ddms:acronym specified and the acronym begins with "USA:", then the organization value must be defined by the USAgency CES. Human Readable: Utilized agency acronyms beginning with "USA:" must be defined in the USAgency CES.

Code Description

For ddms:organization with @ddms:acronym specified, if @ddms:acronym begins with 'USA:', then the country value must be defined by the USAgency CES.

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="IRM-ID-00101">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:organization[@ddms:acronym]">
        <sch:assert test="if (starts-with(normalize-space(@ddms:acronym),'USA:')) then (some $token in $USAgencyAcronymList satisfies $token = normalize-space(substring-
after( @ddms:acronym, ':')) else true()"
            flag="error">[IRM-ID-00101][Error] If element ddms:organization has attribute @ddms:acronym specified and the acronym begins with "USA:", then the
organization value must be defined by the USAgency CES. /&gt;</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.74 - .//Rules/IRM_ID_00102.sch

Rule Description

[IRM-ID-00102][Error] If element ddms:organization has attribute @ddms:acronym specified and the acronym contains a country component that is not "USA:", then the country value must be defined by the GENC CES. Human Readable: Utilized agency acronyms with a country component must have the country defined in the GENC CES.

Code Description

For ddms:organization with @ddms:acronym specified, if @ddms:acronym contains a country component which is not 'USA', then the country value must be defined by the GENC CES.

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="IRM-ID-00102">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:organization[@ddms:acronym]">
        <sch:let name="CC" value="tokenize(normalize-space(@ddms:acronym),':')[1]"/>
        <sch:assert test="if ((count(tokenize(normalize-space(@ddms:acronym),':')) > 1) and not (matches( $CC, 'USA')) ) then (some $token in $gencCountryCodeList satisfies
$token = normalize-space($CC)) else true()"
            flag="error">[IRM-ID-00102][Error] If element ddms:organization has attribute @ddms:acronym specified and the acronym contains a country component that is
not "USA:", then the country value must be defined by the GENC CES. /></sch:assert>
    </sch:rule>
</sch:pattern>
```

2.75 - ../Rules/IRM_ID_00103.sch

Rule Description

[IRM-ID-00103][Error] If element ddms:organization has attribute @ddms:acronym specified, then attribute @ddms:acronym must be of type NmToken.

Code Description

For ddms:organization that have @ddms:acronym, @ddms:acronym must be of type NmToken.

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="IRM-ID-00103">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:organization[@ddms:acronym]">
        <sch:assert test="util:meetsType(@ddms:acronym, $NmTokenPattern)" flag="error">[IRM-ID-00103][Error] If element ddms:organization has attribute @ddms:acronym specified,
then attribute @ddms:acronym must be of type NmToken.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.76 - ../Rules/IRM_ID_00104.sch

Rule Description

[IRM-ID-00104][Error] If element ddms:organization has attribute @ddms:acronym specified and @ddms:acronym has a country prefix, then the agency suffix after the colon delimiter must be of type NmToken.

Code Description

For ddms:organization that have @ddms:acronym with a country prefix, the agency suffix after the colon delimiter must be of type NmToken.

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="IRM-ID-00104">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:organization[@ddms:acronym]">
        <sch:assert test="if (count(tokenize(normalize-space(@ddms:acronym),':')) > 1) then util:meetsType(substring-after(@ddms:acronym,':'), $NmTokenPattern) else true()"
            flag="error">[IRM-ID-00104][Error] If element ddms:organization has attribute @ddms:acronym specified and @ddms:acronym has a country prefix, then the
agency suffix after the colon delimiter must be of type NmToken.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.77 - .//Rules/IRM_ID_00105.sch

Rule Description

[IRM-ID-00105][Error] Use of the @ddms:qualifier on the ddms:nonStateActor element is required.

Code Description

If the ddms:nonStateActor element is specified within a DDMS assertion affected by an IRM assertion, then verify that the @ddms:qualifier attribute is present on the 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="IRM-ID-00105">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:nonStateActor">
        <sch:assert test="@ddms:qualifier" flag="error">[IRM-ID-00105][Error] Use of the @ddms:qualifier on the ddms:nonStateActor element is required.</sch:assert>
    </sch:rule>
</sch:pattern>
```

Chapter 3 - Abstract Patterns

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

3.1 - ./Lib/CompareDateTimes.sch

Code Description

\$context := an xpath to an element

\$primaryDate := an xpath, relative to \$context, to a date to compare against all dates in \$secondaryDateList

\$secondaryDateList := a list of xpaths, relative to \$context, each to a dates in which to compare against \$primaryDate

First, we make sure that the primaryDate is an allowable date format. If the primary date is not a valid date format, then we return true because we cannot guarantee the value provided is not allowed. Then, for each date in \$secondaryDateList we perform the same check for a valid date format and compare the secondaryDate to the primaryDate. To perform comparisons between dates, we use the comparison operator contained in the param \$operator and make sure that all comparisons between primary and secondary dates 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
  $primaryDate := an xpath, relative to $context, to a date to compare against all dates in $secondaryDateList
  $secondaryDateList := a list of xpaths, relative to $context, each to a dates in which to compare against $primaryDate
  $operator := the equality operator to use for comparing each date in $secondaryDateList to $primaryDate

  First, we make sure that the primaryDate is an allowable date format. If the primary date is not a valid
  date format, then we return true because we cannot guarantee the value provided is not allowed. Then, for
  each date in $secondaryDateList we perform the same check for a valid date format and compare the
  secondaryDate to the primaryDate. To perform comparisons between dates, we use the comparison operator
  contained in the param $operator and make sure that all comparisons between primary and secondary dates
  returns true.
-->
<sch:pattern abstract="true" id="CompareDateTimes">
  <sch:rule context="$context">
    <sch:assert test="if ($flag = 'warning' and dtf:isAllowableDateTimeFormat(string($primaryDate))) then every $secondaryDate in $secondaryDateList satisfies
if(dtf:isAllowableDateTimeFormat(string($secondaryDate))) then dtf:compareDateTimeRanges(string($primaryDate), $operator, string($secondaryDate)) else true() else true()"
      flag="warning">
      <sch:value-of select="$ruleText"/>
    </sch:assert>
    <sch:assert test="if ($flag = 'error' and dtf:isAllowableDateTimeFormat(string($primaryDate))) then every $secondaryDate in $secondaryDateList satisfies
if(dtf:isAllowableDateTimeFormat(string($secondaryDate))) then dtf:compareDateTimeRanges(string($primaryDate), $operator, string($secondaryDate)) else true() else true()"
      flag="error">
      <sch:value-of select="$ruleText"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

3.2 - ./Lib/ICIdentifierRestrictions.sch

Code Description

A valid IC-Identifier must meet the following criteria: (1) The id must begin with 'guide:/' (2) The prefix for the id is an integer up to 16 digits with no leading zeros allowed (3) The suffix is an alphanumeric string limited to 36 characters of the set that includes A-Z, a-z, 0-9, underscore, hyphen, and period (4) There are no additional characters proceeding the ID. In order to determine the provided IC-Identifier meets these criteria, the value parameter is matched against the following regex: ^guide://([1-9][0-9]{0,15}|0)/[A-Za-z0-9_-\.\.]{1,36}\$.

Schematron Code

```
<sch:pattern abstract="true" id="ICIdentifierRestrictions">
  <sch:rule context="$context">
    <sch:let name="icidRegEx"
      value="'^guide://([1-9][0-9]{0,15}|0)/[A-Za-z0-9_-\.\.]{1,36}$'"/>
    <sch:assert test="matches(string($value),$icidRegEx)" flag="error">
      <sch:value-of select="$errorMessage"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

3.3 - ../Lib/IsmEnforcement.sch

Code Description

\$qualifier := the qualifier value that requires ism to be present

\$errMsg := the error message text to display when the assertion fails

This abstract pattern checks that if a particular qualifier is specified on a ddms:type element that ism:classification is also specified.

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 that if a particular qualifier is specified on a
ddms:type element that ism:classification is also specified.

$qualifier    := the qualifier value that requires ism to be present
$errMsg       := the error message text to display when the assertion fails

Example usage:
<sch:pattern is-a="DdmsTypeIsmEnforcement" id="IRM_ID_00039" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
<sch:param name="ruleText" value="" />
<sch:param name="codeDesc" value="" />
<sch:param name="context" value="ddms:type[@ddms:qualifier=$qualifier]" />
<sch:param name="errMsg" value=" '
[IRM-ID-00039][Error]
If ddms:type is specified with a qualifier of urn:us:gov:ic:irm:productline then
ism:classification must also be specified.
' "/>
</sch:pattern>

Note: $iso4217TrigraphList is defined in the main document, IRM_XML.xml.
-->
<sch:pattern abstract="true" id="IsmEnforcement">
  <sch:rule context="$context">
    <sch:assert test="@ism:classification" flag="error">
      <sch:value-of select="$errMsg" />
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

3.4 - ./Lib/ValidateValueExistenceInList.sch

Code Description

\$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

This abstract pattern checks to see if an attribute of an element exists in a list.

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="ValidateValueExistenceInList" id="IRM_ID_00027" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:param name="context" value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:language"/>
        <sch:param name="searchTerm" value="@ddms:qualifier"/>
        <sch:param name="list" value="$compoundLanguageQualifierTypeList"/>
        <sch:param name="errMsg" value=""
          [IRM-ID-00034][Error] For element ddms:language, attribute ddms:qualifier must have a
          value in CVEnumIRMCompoundLanguageQualifierType.xml.'" />
      </sch:pattern>

      Note: $iso4217TrigraphList is defined in the main document, IRM_XML.xml.
-->
<sch:pattern abstract="true" id="ValidateValueExistenceInList">
  <sch:rule context="$context">
    <sch:assert test="some $token in $list satisfies $token = $searchTerm or matches($searchTerm, concat('^',$token,'$'))"
      flag="error">
      <sch:value-of select="$errMsg"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

Chapter 4 - Min Accessible Rules

All of the numbered Rules for IRM that are specifically required for the MIN_ACCESSIBLE compliesWith mode are listed in this section. These rules are also enforced when the compliesWith mode is set to MIN_DISCOVERABLE since that is a super set. There are other rules that could come into play when using MIN_ACCESSIBLE if you add any data that would trigger a rule. This set of rules is called out to help understand what MIN_ACCESSIBLE means. These rules may depend on patterns defined in the Abstract Patterns section or on variables defined in the Schematron Schema section.

4.1 - ../Rules/IRM_ID_00062.sch

Rule Description

[IRM-ID-00062][Error] The value of an IC-ID identifier must follow standardized convention. Human Readable: The IC-ID identifier value has to follow standardized convention.

Code Description

This rule uses an abstract pattern that contains the logic for ensuring the value found if a given context exists, both provided as parameters from this implementation, follows the IC-ID identifier standardized convention.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00062" is-a="ICIdentifierRestrictions">
    <sch:param name="context"
        value="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource//ddms:identifier[@ddms:qualifier='IC-ID']"/>
    <sch:param name="value" value="string(@ddms:value)"/>
    <sch:param name="errorMessage"
        value="'[IRM-ID-00062][Error] The value of an IC-ID identifier must follow standardized convention. Human Readable: The IC-ID identifier value has to follow
standardized convention.'"/>
</sch:pattern>
```

4.2 - ../Rules/IRM_ID_00063.sch

Rule Description

[IRM-ID-00063][Error] Element ddms:resource/ddms:creator/ddms:organization must specify attribute @ddms:acronym. Human Readable: The DDMS card must specify a creator organization with an IC agency acronym for the referenced resource.

Code Description

We make sure that element ddms:resource/ddms:creator/ddms:organization exists and specifies attribute @ddms:acronym.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00063">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource">
        <sch:assert test="ddms:creator/ddms:organization/@ddms:acronym" flag="error">[IRM-ID-00063][Error] Element ddms:resource/ddms:creator/ddms:organization must specify
attribute @ddms:acronym. Human Readable: The DDMS card must specify a creator organization with an IC agency acronym for the referenced resource.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.3 - ../Rules/IRM_ID_00064.sch

Rule Description

[IRM-ID-00064][Error] Element ddms:resource/ddms:dates must specify attribute @ddms:created. Human Readable: The DDMS card must specify the date on which the referenced resource was created.

Code Description

We make sure that element ddms:resource/ddms:dates exists and specifies attribute @ddms:created.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00064">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource">
        <sch:assert test="ddms:dates/@ddms:created" flag="error">[IRM-ID-00064][Error] Element ddms:resource/ddms:dates must specify attribute @ddms:created. Human Readable:
The DDMS card must specify the date on which the referenced resource was created.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.4 - ../Rules/IRM_ID_00065.sch

Rule Description

[IRM-ID-00065][Error] Attribute ddms:resource/ddms:dates/@ddms:created must be castable as an xs:dateTime type. Human Readable: The date on which the referenced resource was created must be a dateTime type.

Code Description

We make sure that attribute dms:resource/ddms:dates/@ddms:created is castable as an xs:dateTime type.

Schematron Code

```
<?ICEA pattern?>
<?ICEA min_accessible?>
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<sch:pattern id="IRM-ID-00065">
    <sch:rule context="tdf:*[tdf:Assertion//irm:ICResourceMetadataPackage]/tdf:Assertion/tdf:StructuredStatement/ddms:resource/ddms:dates[@ddms:created]">
        <sch:assert test="@ddms:created castable as xs:dateTime" flag="error">[IRM-ID-00065][Error] Attribute ddms:resource/ddms:dates/@ddms:created must be castable as an
xs:dateTime type. Human Readable: The date on which the referenced resource was created must be a dateTime type.</sch:assert>
    </sch:rule>
</sch:pattern>
```

Chapter 5 - Schematron Schema

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

5.1 - ../IRM_XML.sch

Code Description

This is the root file for the IRM Schematron ruleset. It loads all of the required CVEs, declares some variables, and includes all of the Rule .sch files.

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">
    <sch:ns uri="http://www.w3.org/2001/XMLSchema" prefix="xsd"/>
    <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:ntk" prefix="ntk"/>
    <sch:ns uri="urn:us:gov:ic:tdf" prefix="tdf"/>
    <sch:ns uri="urn:us:gov:ic:id" prefix="icid"/>
    <sch:ns uri="urn:us:gov:ic:usagency" prefix="usagency"/>
    <sch:ns uri="urn:us:gov:ic:pm" prefix="pm"/>
    <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:icgenc" prefix="genc"/>
    <sch:ns uri="urn:us:mil:ces:metadata:ddms:5" prefix="ddms"/>
    <sch:ns uri="urn:us:gov:ic:cve" prefix="cve"/>
    <sch:ns uri="http://www.w3.org/1999/xlink" prefix="xlink"/>
    <sch:ns uri="http://www.w3.org/1999/XSL/Transform" prefix="xsl"/>
    <sch:ns uri="date:time:function" prefix="dtf"/>
    <sch:ns prefix="util" uri="urn:us:gov:ic:irm:xsl:util"/>
    <sch:let name="IRM_COMPLIES_WITH" value="ddms:resource/@ddms:compliesWith"/>
    <sch:let name="MIN_DISCOVERABLE_OR_GREATER"
        value="util:containsAnyOfTheTokens($IRM_COMPLIES_WITH, ('MIN_DISCOVERABLE'))"/>
    <sch:let name="MIN_ACCESSIBLE_OR_GREATER"
        value="$MIN_DISCOVERABLE_OR_GREATER or util:containsAnyOfTheTokens($IRM_COMPLIES_WITH, ('MIN_ACCESSIBLE'))"/>
    <!-- (U) Resources -->
<sch:let name="coverageIso3166DigraphList"
    value="document('.../CVE/IRM/CVEnumIRMCoverageISO3166Digraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
    <sch:let name="iso639DigraphList"
    value="document('.../CVE/IRM/CVEnumIRMISO639Digraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
    <sch:let name="iso639-2TrigraphList"
    value="document('.../CVE/IRM/CVEnumIRMISO639-2Trigraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
    <sch:let name="iso639-3TrigraphList"
    value="document('.../CVE/IRM/CVEnumIRMISO639-3Trigraph.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
    <sch:let name="mimeTypeList"
    value="document('.../CVE/MIME/CVEnumMIMETYPE.xml')//cve:Value"/>
    <sch:let name="deprecatedMimeTypeList"
    value="document('.../CVE/MIME/CVEnumMIMETYPE.xml')/cve:CVE/cve:Enumeration/cve:Term[./@deprecated]/cve:Value"/>
    <sch:let name="compoundLanguageQualifierTypeList"
    value="document('.../CVE/IRM/CVEnumIRMCompoundLanguageQualifierType.xml')//cve:Value"/>
    <sch:let name="intelDisciplineComponentTechniqueList"
    value="document('.../CVE/INTDIS/CVEnumIntelDisciplineComponentTechnique.xml')//cve:Value"/>
    <sch:let name="intelDisciplineComponentList"
    value="document('.../CVE/INTDIS/CVEnumIntelDisciplineComponent.xml')//cve:Value"/>
    <sch:let name="intelDisciplineList"
    value="document('.../CVE/INTDIS/CVEnumIntelDiscipline.xml')//cve:Value"/>
    <sch:let name="positiveIntelList"
    value="document('.../CVE/IRM/CVEnumIRMPositiveIntel.xml')//cve:Value"/>
    <sch:let name="productionMetricsSubjectList"
    value="document('.../CVE/PM/CVEnumPMSubject.xml')//cve:Value"/>
```

```
<sch:let name="productionMetricsCoverageList"
  value="document('.../CVE/PM/CVEnumPMCoverage.xml')//cve:Value"/>
<sch:let name="activityList"
  value="document('.../CVE/IRM/CVEnumIRMActivity.xml')//cve:Value"/>
<sch:let name="executableIndicatorList"
  value="document('.../CVE/IRM/CVEnumIRMExecutableIndicator.xml')//cve:Value"/>
<sch:let name="maliciousCodeIndicatorList"
  value="document('.../CVE/IRM/CVEnumIRMMaliciousCodeIndicator.xml')//cve:Value"/>
<sch:let name="coveragePrecedenceList"
  value="document('.../CVE/IRM/CVEnumIRMCoveragePrecedence.xml')//cve:Value"/>
<sch:let name="USAgencyAcronymList"
  value="document('.../CVE/USAgency/CVEnumUSAgencyAcronym.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="nonStateActorsList"
  value="document('.../CVE/PM/CVEnumPMNonStateActors.xml')//cve:Value"/>
<sch:let name="gencCountryCodeList"
  value="document('.../CVE/IC-GENC/CVEnumGENCCountryCode.xml')//cve:Value"/>
<sch:let name="gencSubDivisionList"
  value="document('.../CVE/IC-GENC/CVEnumGENCSubDivisionCode.xml')//cve:Value"/>
<!-- ***** --><!-- * 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'"/>
<!-- ***** --><!-- * Abstract Rule and Pattern Includes * --><!-- ***** -->
<sch:include href="./Lib/ValidateValueExistenceInList.sch"/>
<sch:include href="./Lib/DateListYearRangeRule.sch"/>
<sch:include href="./Lib/DateYearRangeRule.sch"/>
<sch:include href="./Lib/CompareDateTimes.sch"/>
<sch:include href="./Lib/IsmEnforcement.sch"/>
<sch:include href="./Lib/ICIdentifierRestrictions.sch"/>
<sch:include href="./Lib/TypeConstraintPatterns.sch"/>
<!-- ***** --><!-- * Custom XSLT2 Function Definitions * --><!-- ***** -->
  Returns true if the raw value match the provided regular expressions.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  name="util:meetsType"
  as="xs:boolean">
  <xsl:param name="value"/>
  <xsl:param name="typePattern" as="xs:string"/>
  <xsl:value-of select="matches(string($value), concat('^(', $typePattern, ')$'))"/>
</xsl:function>
<!--
  Returns true if any token in the attribute value matches at least one token in the provided list.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  name="util:containsAnyOfTheTokens"
  as="xs:boolean">
  <xsl:param name="attribute"/>
  <xsl:param name="tokenList" as="xs:string+"/>
  <xsl:value-of select="some $attrToken in tokenize(normalize-space(string($attribute)), ' ') satisfies $attrToken = $tokenList"/>
</xsl:function>
<!--
```

```

    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})?${}'"/>
        <!-- year, month, day, hour, minute, seconds, optional milliseconds -->
<xsl:variable name="YYYY-MM-DDThh-mm-ss"
              as="xs:string"
              select="'^{\d{4}}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d{1,3})?(Z|[\+-]\d{2}:\d{2})?${}'"/>
        <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 strictly 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 strictly 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 < $secondaryStart"/>
</xsl:function>
<!--
Determines if the date range implied by the string representation in
primary is strictly after the date range implied by the string
representation in secondary. Returns true if the end of the date
range implied by primary is less than the start of the date range
implied by secondary; otherwise, returns false.
@param {xs:string} primary String representation of a date in one
    of the allowable formats.
@param {xs:string} secondary String representation of a date in one
    of the allowable formats.
@returns {xs:boolean} Returns true if the date range implied by primary
    is strictly later than the date range implied by secondary; otherwise,
    returns false.
-->
<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              name="dtf:isAfter"
              as="xs:boolean">
    <xsl:param name="primary" as="xs:string"/>
    <xsl:param name="secondary" as="xs:string"/>
    <xsl:variable name="primaryStart"
                  as="xs:dateTime"
                  select="dtf:startDate($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 $operator = '<') and (($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 $operator = '>') and (($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 = '<' or $operator = '<='">
                    <xsl:value-of select="dtf:isBefore($primary, $secondary) or dtf:overlaps($primary, $secondary)"/>
                </xsl:when>
                <!-- 'Greater Than' and 'Greater Than or Equal' -->
                <xsl:when test="$operator = 'gt' or $operator = '>' or $operator = '>='">
                    <xsl:value-of select="dtf:isAfter($primary, $secondary) or dtf:overlaps($primary, $secondary)"/>
                </xsl:when>
                <!-- Default to false() -->
            <xsl:otherwise>
                <xsl:value-of select="false()"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
        <xsl:value-of select="false()"/>
    </xsl:otherwise>
</xsl:function>

```

```
        <xsl:value-of select="false()"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:function>
  <!--*****--><!-- (U) IRM ID Rules --><!--*****--><!--(U) -->
<sch:include href="./Rules/IRM_ID_00002.sch"/>
<sch:include href="./Rules/IRM_ID_00005.sch"/>
<sch:include href="./Rules/IRM_ID_00006.sch"/>
<sch:include href="./Rules/IRM_ID_00007.sch"/>
<sch:include href="./Rules/IRM_ID_00010.sch"/>
<sch:include href="./Rules/IRM_ID_00015.sch"/>
<sch:include href="./Rules/IRM_ID_00016.sch"/>
<sch:include href="./Rules/IRM_ID_00017.sch"/>
<sch:include href="./Rules/IRM_ID_00019.sch"/>
<sch:include href="./Rules/IRM_ID_00020.sch"/>
<sch:include href="./Rules/IRM_ID_00021.sch"/>
<sch:include href="./Rules/IRM_ID_00022.sch"/>
<sch:include href="./Rules/IRM_ID_00023.sch"/>
<sch:include href="./Rules/IRM_ID_00024.sch"/>
<sch:include href="./Rules/IRM_ID_00025.sch"/>
<sch:include href="./Rules/IRM_ID_00029.sch"/>
<sch:include href="./Rules/IRM_ID_00030.sch"/>
<sch:include href="./Rules/IRM_ID_00033.sch"/>
<sch:include href="./Rules/IRM_ID_00034.sch"/>
<sch:include href="./Rules/IRM_ID_00036.sch"/>
<sch:include href="./Rules/IRM_ID_00040.sch"/>
<sch:include href="./Rules/IRM_ID_00041.sch"/>
<sch:include href="./Rules/IRM_ID_00042.sch"/>
<sch:include href="./Rules/IRM_ID_00043.sch"/>
<sch:include href="./Rules/IRM_ID_00044.sch"/>
<sch:include href="./Rules/IRM_ID_00045.sch"/>
<sch:include href="./Rules/IRM_ID_00046.sch"/>
<sch:include href="./Rules/IRM_ID_00047.sch"/>
<sch:include href="./Rules/IRM_ID_00048.sch"/>
<sch:include href="./Rules/IRM_ID_00050.sch"/>
<sch:include href="./Rules/IRM_ID_00051.sch"/>
<sch:include href="./Rules/IRM_ID_00052.sch"/>
<sch:include href="./Rules/IRM_ID_00053.sch"/>
<sch:include href="./Rules/IRM_ID_00054.sch"/>
<sch:include href="./Rules/IRM_ID_00055.sch"/>
<sch:include href="./Rules/IRM_ID_00059.sch"/>
<sch:include href="./Rules/IRM_ID_00062.sch"/>
<sch:include href="./Rules/IRM_ID_00063.sch"/>
<sch:include href="./Rules/IRM_ID_00064.sch"/>
<sch:include href="./Rules/IRM_ID_00065.sch"/>
<sch:include href="./Rules/IRM_ID_00068.sch"/>
<sch:include href="./Rules/IRM_ID_00070.sch"/>
<sch:include href="./Rules/IRM_ID_00071.sch"/>
<sch:include href="./Rules/IRM_ID_00072.sch"/>
<sch:include href="./Rules/IRM_ID_00073.sch"/>
<sch:include href="./Rules/IRM_ID_00074.sch"/>
<sch:include href="./Rules/IRM_ID_00075.sch"/>
<sch:include href="./Rules/IRM_ID_00076.sch"/>
<sch:include href="./Rules/IRM_ID_00077.sch"/>
```

```
<sch:include href="./Rules/IRM_ID_00078.sch"/>
<sch:include href="./Rules/IRM_ID_00079.sch"/>
<sch:include href="./Rules/IRM_ID_00080.sch"/>
<sch:include href="./Rules/IRM_ID_00081.sch"/>
<sch:include href="./Rules/IRM_ID_00082.sch"/>
<sch:include href="./Rules/IRM_ID_00083.sch"/>
<sch:include href="./Rules/IRM_ID_00084.sch"/>
<sch:include href="./Rules/IRM_ID_00085.sch"/>
<sch:include href="./Rules/IRM_ID_00086.sch"/>
<sch:include href="./Rules/IRM_ID_00087.sch"/>
<sch:include href="./Rules/IRM_ID_00088.sch"/>
<sch:include href="./Rules/IRM_ID_00089.sch"/>
<sch:include href="./Rules/IRM_ID_00090.sch"/>
<sch:include href="./Rules/IRM_ID_00091.sch"/>
<sch:include href="./Rules/IRM_ID_00092.sch"/>
<sch:include href="./Rules/IRM_ID_00093.sch"/>
<sch:include href="./Rules/IRM_ID_00094.sch"/>
<sch:include href="./Rules/IRM_ID_00095.sch"/>
<sch:include href="./Rules/IRM_ID_00096.sch"/>
<sch:include href="./Rules/IRM_ID_00097.sch"/>
<sch:include href="./Rules/IRM_ID_00098.sch"/>
<sch:include href="./Rules/IRM_ID_00099.sch"/>
<sch:include href="./Rules/IRM_ID_00100.sch"/>
<sch:include href="./Rules/IRM_ID_00101.sch"/>
<sch:include href="./Rules/IRM_ID_00102.sch"/>
<sch:include href="./Rules/IRM_ID_00103.sch"/>
<sch:include href="./Rules/IRM_ID_00104.sch"/>
<sch:include href="./Rules/IRM_ID_00105.sch"/>
</sch:schema>
```

Chapter 6 - Removed Rules

All of the numbered Rules for IRM 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.

6.1 - `./Rules/deleted/IRM_ID_00001.sch`

Rule Description

[IRM-ID-00001] Removed in V9 because CVEnumIRMCoverageFIPSDigraph.xml was no longer supported. Human Readable: Removed in V9.

6.2 - `./Rules/deleted/IRM_ID_00003.sch`

Rule Description

[IRM-ID-00003] Removed in V9. Human Readable: Removed in V9.

6.3 - `./Rules/deleted/IRM_ID_00004.sch`

Rule Description

[IRM-ID-00004] Removed in V4. Human Readable: Removed in V4.

6.4 - `./Rules/deleted/IRM_ID_00008.sch`

Rule Description

[IRM-ID-00008] Removed in 2016MAY.

6.5 - `./Rules/deleted/IRM_ID_00009.sch`

Rule Description

[IRM-ID-00009] Removed in 2016MAY.

6.6 - `./Rules/deleted/IRM_ID_00011.sch`

Rule Description

[IRM-ID-00011] Rule removed in V8 because it is covered by rule IRM-ID-00012. Human Readable: Rule removed in V8.

6.7 - `./Rules/deleted/IRM_ID_00012.sch`

Rule Description

[IRM-ID-00012] Removed in V9. Human Readable: Removed in V9.

6.8 - **./Rules/deleted/IRM_ID_00013.sch**

Rule Description

[IRM-ID-00013] Rule removed in V8 because it is covered by rule IRM-ID-00014. Human Readable: Removed in V8.

6.9 - **./Rules/deleted/IRM_ID_00014.sch**

Rule Description

[IRM-ID-00014] Removed in V9. Human Readable: Removed in V9.

6.10 - **./Rules/deleted/IRM_ID_00018.sch**

Rule Description

[IRM-ID-00018] Rule removed in V7.

6.11 - **./Rules/deleted/IRM_ID_00026.sch**

Rule Description

[IRM-ID-00026] Removed in V4. Human Readable: Removed in V4.

6.12 - **./Rules/deleted/IRM_ID_00027.sch**

Rule Description

[IRM-ID-00027] Removed in V6. Human Readable: Removed in V6.

6.13 - **./Rules/deleted/IRM_ID_00028.sch**

Rule Description

[IRM-ID-00028] Removed in V6. Human Readable: Removed in V6.

6.14 - **./Rules/deleted/IRM_ID_00031.sch**

Rule Description

[IRM-ID-00031] Removed in 2016MAY.

6.15 - **./Rules/deleted/IRM_ID_00032.sch**

Rule Description

[IRM-ID-00032][] Removed in V6. Human Readable: Removed in V6.

6.16 - **./Rules/deleted/IRM_ID_00035.sch**

Rule Description

[IRM-ID-00035][] Rule removed in V9. Human Readable: Rule removed in V9.

6.17 - **./Rules/deleted/IRM_ID_00037.sch**

Rule Description

[IRM-ID-00037][] Removed in V9. Human Readable: Rule removed in V9.

6.18 - **./Rules/deleted/IRM_ID_00038.sch**

Rule Description

[IRM-ID-00038][] Removed in V9. Human Readable: Rule removed in V9.

6.19 - **./Rules/deleted/IRM_ID_00039.sch**

Rule Description

[IRM-ID-00039][] Rule removed in V9. Human readable: Rule removed in V9.

6.20 - **./Rules/deleted/IRM_ID_00049.sch**

Rule Description

[IRM-ID-00049][] Removed in 2016MAY, replaced by IRM-ID-00090

6.21 - **./Rules/deleted/IRM_ID_00056.sch**

Rule Description

[IRM-ID-00056][] Rule removed in V9. Human Readable: Rule removed in V9.

6.22 - **./Rules/deleted/IRM_ID_00057.sch**

Rule Description

[IRM-ID-00057] Rule introduced in V8 and removed during V8 CR adjudication. Never part of a signed release. Human readable: Rule removed during V8 CR adjudication.

6.23 - **./Rules/deleted/IRM_ID_00058.sch**

Rule Description

[IRM-ID-00058] Rule introduced in V8 and removed during V8 CR adjudication. Never part of a signed release. Human Readable: Rule removed during V8 CR adjudication.

6.24 - **./Rules/deleted/IRM_ID_00060.sch**

Rule Description

[IRM-ID-00060] Rule introduced in V8 and removed during V8 CR adjudication. Never part of a signed release. Human Readable: Rule removed during V8 CR adjudication.

6.25 - **./Rules/deleted/IRM_ID_00061.sch**

Rule Description

[IRM-ID-00061] Rule removed in V9. Human Readable: Rule removed in V9.

6.26 - **./Rules/deleted/IRM_ID_00066.sch**

Rule Description

[IRM-ID-00066] Removed in V9. Human Readable: Rule removed in V9.

6.27 - **./Rules/deleted/IRM_ID_00067.sch**

Rule Description

[IRM-ID-00067] Removed in V9. Human Readable: Rule removed in V9.

6.28 - **./Rules/deleted/IRM_ID_00069.sch**

Rule Description

[IRM-ID-00069] Rule removed in V9 because the otherNetwork attribute was removed. Networks Names are now controlled through the VIRT CVE, which uses a regex for other network names. Human Readable: Rule removed in V9.