Guide to Schematron Rules and Patterns

IC-EDH Schematron Guide

Version 2019-MAR

December 1, 2022

Table of Contents

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for IC-EDH. This guide is generated from the IC-EDH 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 IC-EDH. 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 IC-EDH. The abstract patterns may be used in numbered rules or provided as reference for use in rules developed by users of IC-EDH. Each abstract pattern has a code description and a code block with the abstract Schematron pattern.

Chapter 4 is a listing of the master IC-EDH 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 IC-EDH are encoded using ISO Schematron. Schematron is a rule-based validation language that uses XML Path Language to make assertions about an XML document.

IC-EDH 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 IC-EDH's Schematron rules.

Chapter 2 - Rules

All of the numbered Rules for IC-EDH 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 IC-EDH-ID-XXXXX, with rule files named IC-EDH_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/IC-EDH_ID_00001.sch

Rule Description

[IC-EDH-ID-00001][Error] Every attribute in the EDH namespace must have a non-whitespace value. Human Readable: All attributes in the EDH namespace must specify a value.

Code Description

Make sure any attribute in the EDH namespace has a value if it is present.

Schematron Code

```xml
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
        This document has been approved for Public Release and is available for use without restriction.
      -->

<sch:pattern id="IC-EDH-ID-00001">
            <sch:rule id="IC-EDH-ID-00001-R1" context="*[@edh:*]">
                <sch:assert test="every $attribute in @edh:* satisfies normalize-space(string($attribute))"
                            flag="error"
                            role="error">[IC-EDH-ID-00001][Error] Every attribute in the EDH namespace must have a non-whitespace value. Human Readable: All attributes in the EDH
namespace must specify a value.</sch:assert>
            </sch:rule>
        </sch:pattern>
```

2.2 - .//Rules/IC-EDH_ID_00002.sch

Rule Description

[IC-EDH-ID-00002][Error] The attribute DESVersion in the namespace urn:us:gov:ic:edh must be specified. Human Readable: The data encoding specification version number must be specified.

Code Description

Make sure that the attribute edh:DESVersion is specified.

Schematron Code

```xml
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
          This document has been approved for Public Release and is available for use without restriction.
       -->

<sch:pattern id="IC-EDH-ID-00002">
                <sch:rule id="IC-EDH-ID-00002-R1" context="/">
                    <sch:assert test="some $element in descendant-or-self::node() satisfies $element/@edh:DESVersion"
                                flag="error"
                                role="error">[IC-EDH-ID-00002][Error] The attribute DESVersion in the namespace urn:us:gov:ic:edh must be specified. Human Readable: The data encoding
specification version must be specified.</sch:assert>
                </sch:rule>
            </sch:pattern>
```

2.3 - .//Rules/IC-EDH_ID_00003.sch

Rule Description

[IC-EDH-ID-00003][Error] The dataItemCreateDateTime must not be later the ism:createDate. Human Readable: Data items cannot be newer than their security markings.

Code Description

For EDH element DataItemCreateDateTime, ensure that the sibling ARH element (Security or ExternalSecurity) has ism:createDate that is not older than the data item itself. This comparison is done strictly on the date, and to compensate for the loss of using time, the ism:createDate is incremented one day and less than, not less than or equal to.

Schematron Code

```xml
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
        This document has been approved for Public Release and is available for use without restriction.
     -->

<sch:pattern id="IC-EDH-ID-00003">
            <sch:rule id="IC-EDH-ID-00003-R1"
                    context="edh:DataItemCreateDateTime[following-sibling::arh:*/@ism:createDate]">
                <sch:assert test="xsd:date(substring-before(xsd:string(.),'T')) &lt; functx:next-day(following-sibling::arh:*/@ism:createDate)"
                        flag="error"
                        role="error">[IC-EDH-ID-00003][Error] The dataItemCreateDateTime must not be later the ism:createDate. Human Readable: Data items cannot be newer than their
security markings.</sch:assert>
            </sch:rule>
</sch:pattern>
```

2.4 - .//Rules/IC-EDH_ID_00004.sch

Rule Description

[IC-EDH-ID-00004][Error] The EDH elements cannot be used as root elements. Human Readable: EDH is not designed to stand-alone and therefore should never be used as a root element.

Code Description

This rule is to ensure edh:Edh or edh:ExternalEdh are not used as the root 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="IC-EDH-ID-00004">
            <sch:rule id="IC-EDH-ID-00004-R1" context="/edh:*">
                <sch:assert test="false()" flag="error" role="error">[IC-EDH-ID-00004][Error] The EDH elements cannot be used as root elements. Human Readable: EDH is not designed to
stand-alone and therefore should never be used as a root element.</sch:assert>
            </sch:rule>
        </sch:pattern>
```

2.5 - .//Rules/IC-EDH_ID_00005.sch

Rule Description

[IC-EDH-ID-00005][Error] Every element in the EDH namespace must have a non-whitespace value. Human Readable: All elements in the EDH namespace must specify a value.

Code Description

Make sure any element in the EDH namespace has a value if it is present.

Schematron Code

```xml
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
          This document has been approved for Public Release and is available for use without restriction.
       -->

<sch:pattern id="IC-EDH-ID-00005">
              <sch:rule id="IC-EDH-ID-00005-R1" context="edh:*">
                <sch:assert test="normalize-space(.)" flag="error" role="error">[IC-EDH-ID-00005][Error] Every element in the EDH namespace must have a non-whitespace value. Human
Readable: All elements in the EDH namespace must specify a value.</sch:assert>
              </sch:rule>
          </sch:pattern>
```

2.6 - .//Rules/IC-EDH_ID_00006.sch

Rule Description

[IC-EDH-ID-00006][Error] If the Country of the ResponsibleEntity is [USA] then the value of Organization must be a term from the USAgency CES. Human Readable: If the Country element contains USA, the agency in the Organization element must be defined in 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 is-a="ValidateValueExistenceInList" id="IC-EDH-ID-00006">
                <sch:param name="context" value="edh:Country[.='USA']"/>
                <sch:param name="list" value="$usagencyList"/>
                <sch:param name="searchTerm" value="following-sibling::edh:Organization"/>
                <sch:param name="errMsg"
                        value="' [IC-EDH-ID-00006][Error] If the Country of the ResponsibleEntity is [USA] then the value of Organization must be a term from the USAgency CES. Human
Readable: If the Country element contains USA, the agency in the Organization element must be defined in the USAgency CES. '"/>
              </sch:pattern>
```

2.7 - .//Rules/IC-EDH_ID_00008.sch

Rule Description

[IC-EDH-ID-00008][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'ISM' that is version '201903' (Version:2019-MAR) or later. NOTE: This is not an error of the instance document but of the validation environment itself.

Code Description

This rule uses an abstract pattern to consolidate logic. It verifies that the validation infrastructure is using the version specified in parameters.

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="IC-EDH-ID-00008" is-a="ValidateValidationEnvCVE">
                <sch:param name="MinVersion" value="'201903'"/>
                <sch:param name="SpecToCheck" value="'ISM'"/>
                <sch:param name="pathToDocument"
                        value="'../../CVE/ISM/CVEnumISMClassificationAll.xml'"/>
                <sch:param name="RuleID" value="'IC-EDH-ID-00008'"/>
        </sch:pattern>
```

2.8 - .//Rules/IC-EDH_ID_00011.sch

Rule Description

[IC-EDH-ID-00011][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'IC-ID' that is version '1' or later. NOTE: This is not an error of the instance document but of the validation environment itself.

Code Description

This rule uses an abstract pattern to consolidate logic. It verifies that the validation infrastructure is using the version specified in parameters.

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="IC-EDH-ID-00011" is-a="ValidateValidationEnvSchema">
                <sch:param name="MinVersion" value="'1'"/>
                <sch:param name="SpecToCheck" value="'IC-ID'"/>
                <sch:param name="pathToDocument" value="'../../Schema/IC-ID/IC-ID.xsd'"/>
                <sch:param name="RuleID" value="'IC-EDH-ID-00011'"/>
            </sch:pattern>
```

2.9 - .//Rules/IC-EDH_ID_00012.sch

Rule Description

[IC-EDH-ID-00012][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'USAgency' that is version '201703.201802' (Version:2017-MAR Revision:2018-FEB) or later. NOTE: This is not an error of the instance document but of the validation environment itself.

Code Description

This rule uses an abstract pattern to consolidate logic. It verifies that the validation infrastructure is using the version specified in parameters.

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="IC-EDH-ID-00012" is-a="ValidateValidationEnvCVE">
                <sch:param name="MinVersion" value="'201703.201802'"/>
                <sch:param name="SpecToCheck" value="'USAgency'"/>
                <sch:param name="pathToDocument"
                        value="'../../CVE/USAgency/CVEnumUSAgencyAcronym.xml'"/>
                <sch:param name="RuleID" value="'IC-EDH-ID-00012'"/>
        </sch:pattern>
```

2.10 - .//Rules/IC-EDH_ID_00013.sch

Rule Description

[IC-EDH-ID-00013][Error] Element edh:DataItemCreateDateTime of type xs:dateTime must have a timezone. Human Readable: The EDH element DataItemCreateDateTime must have a timezone.

Code Description

The EDH element edh:DataItemCreateDateTime must have a timezone. According to http://www.w3.org/TR/xmlschema-2/#dateTime, datetime is represented by: '-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)? where the timezone zzzzzz is represented by: (('+' | '-') hh ':' mm) | 'Z' This rule enforces and makes the timezone zzzzzz mandatory.

Schematron Code

```xml
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
        This document has been approved for Public Release and is available for use without restriction.
     -->

<sch:pattern id="IC-EDH-ID-00013">
                <sch:rule id="IC-EDH-ID-00013-R1" context="edh:DataItemCreateDateTime">
                    <sch:assert test="matches(., '^-?\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?(((\+|-)\d{2}:\d{2})|Z)$')"
                                flag="error"
                                role="error">[IC-EDH-ID-00013][Error] edh:DataItemCreateDateTime of type xs:dateTime must have a timezone. Human Readable: The EDH element
DataItemCreateDateTime must have a timezone.</sch:assert>
                </sch:rule>
            </sch:pattern>
```

2.11 - .//Rules/IC-EDH_ID_00014.sch

Rule Description

[IC-EDH-ID-00014][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'ISMCAT' that is version '201903' (Version:2019-MAR) or later. NOTE: This is not an error of the instance document but of the validation environment itself.

Code Description

This rule uses an abstract pattern to consolidate logic. It verifies that the validation infrastructure is using the version specified in parameters.

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="IC-EDH-ID-00014" is-a="ValidateValidationEnvCVE">
                <sch:param name="MinVersion" value="'201903'"/>
                <sch:param name="SpecToCheck" value="'ISMCAT'"/>
                <sch:param name="pathToDocument"
                        value="'../../CVE/ISMCAT/CVEnumISMCATTetragraph.xml'"/>
                <sch:param name="RuleID" value="'IC-EDH-ID-00014'"/>
        </sch:pattern>
```

2.12 - .//Rules/IC-EDH_ID_00015.sch

Rule Description

[IC-EDH-ID-00015][Error] There must be one and only one edh:ResponsibleEntity element where role="Custodian" There may be zero or one edh:ResponsibleEntity element where role="Originator" Human Readable: There must be only one and only one edh:ResponsibleEntity element with the role "Custodian." Additionally, there may be zero or one edh:ResponsibleEntity element with the role "Originator."

Code Description

Schematron Code

```xml
<?ICEA pattern?>
<!-- Notices - Distribution Notice:
        This document has been approved for Public Release and is available for use without restriction.
    -->

<sch:pattern id="IC-EDH-ID-00015">
                <sch:rule id="IC-EDH-ID-00015-R1" context="edh:Edh">
                    <sch:assert test="count(./edh:ResponsibleEntity[@edh:role='Custodian']) = 1"
                            flag="error"
                            role="error">[IC-EDH-ID-00015][Error] There must be one and only one edh:ResponsibleEntity element where role="Custodian"</sch:assert>
                    <sch:assert test="count(./edh:ResponsibleEntity[@edh:role='Originator']) &lt;= 1"
                            flag="error"
                            role="error">[IC-EDH-ID-00015][Error] There may be zero or one edh:ResponsibleEntity element where role="Originator"</sch:assert>
                </sch:rule>
                <sch:rule id="IC-EDH-ID-00015-R2" context="edh:ExternalEdh">
                    <sch:assert test="count(./edh:ResponsibleEntity[@edh:role='Custodian']) = 1"
                            flag="error"
                            role="error">[IC-EDH-ID-00015][Error] There must be one and only one edh:ResponsibleEntity element where role="Custodian"</sch:assert>
                    <sch:assert test="count(./edh:ResponsibleEntity[@edh:role='Originator']) &lt;= 1"
                            flag="error"
                            role="error">[IC-EDH-ID-00015][Error] There may be zero or one edh:ResponsibleEntity element where role="Originator"</sch:assert>
                </sch:rule>
            </sch:pattern>
```

2.13 - .//Rules/IC-EDH_ID_00016.sch

Rule Description

[IC-EDH-ID-00016][Warning] edh:DESVersion attribute SHOULD be specified as version 201903 (Version:2019-MAR) with an optional extension.

Code Description

This rule supports extending the version identifier with an optional trailing hyphen and up to 23 additional characters. The version must match the regular expression "^201903(-.{1,23})?$".

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="IC-EDH-ID-00016">
                <sch:rule id="IC-EDH-ID-00016-R1" context="*[@edh:DESVersion]">
                    <sch:assert test="matches(@edh:DESVersion,'^201903(\-.{1,23})?$')"
                                flag="warning"
                                role="warning">[IC-EDH-ID-00016][Warning] edh:DESVersion attribute SHOULD be specified as version 201903 (Version:2019-MAR) with an optional extension. The
value provided was:
        <sch:value-of select="@edh:DESVersion"/>
                    </sch:assert>
                </sch:rule>
            </sch:pattern>
```

2.14 - .//Rules/IC-EDH_ID_00017.sch

Rule Description

[IC-EDH-ID-00017][Error] The @usagency:CESVersion is missing.

Code Description

This rule verifies that the USAgency CESVersion exist.

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="IC-EDH-ID-00017">
            <sch:rule id="IC-EDH-ID-00017-R1" context="/">
              <sch:assert test="//@usagency:CESVersion" flag="error" role="error">[IC-EDH-ID-00017][Error] The @usagency:CESVersion is missing.</sch:assert>
            </sch:rule>
        </sch:pattern>
```

2.15 - .//Rules/IC-EDH_ID_00018.sch

Rule Description

[IC-EDH-ID-00018][Error] The ism:ISMCATCESVersion is missing.

Code Description

This rule verifies that the ISMCAT CESVersion exist.

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="IC-EDH-ID-00018">
              <sch:rule id="IC-EDH-ID-00018-R1" context="/">
                <sch:assert test="//@ism:ISMCATCESVersion" flag="error" role="error">[IC-EDH-ID-00018][Error] The ism:ISMCATCESVersion is missing.</sch:assert>
              </sch:rule>
       </sch:pattern>
```

Chapter 3 - Abstract Patterns

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

3.1 - .//Lib/ValidateValidationEnvCVE.sch

Code Description

This abstract pattern checks to see if the validation environment has at least the version / revision of the CVE as of the writing of this specification. The calling rule must pass in $MinVersion, $SpecToCheck, $pathToDocument, $RuleID.

Schematron Code

```xml
<!--
    This abstract pattern checks to see the version of a CVE is greater than or equal to a passed in parameter.

    $MinVersion             := the version that SpecToCheck must be equal to or greater than.
    $SpecToCheck            := Name the spec whose version in the infrastructure is being checked.
    $pathToDocument         := Relative path to the document cve that has ther version string
    $RuleID                 := The number of the rule in the concrete file.


-->

<sch:pattern xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                        abstract="true"
                        id="ValidateValidationEnvCVE">
            <sch:rule id="ValidateValidationEnvCVE-R1" context="/">
                <sch:assert test="document($pathToDocument)//cve:CVE//@specVersion castable as xs:double and document($pathToDocument)//cve:CVE//@specVersion &gt;= $MinVersion"
                            flag="error"
                            role="error">Version [
        <sch:value-of select="document($pathToDocument)//cve:CVE//@specVersion"/>] of
        <sch:value-of select="$SpecToCheck"/>found; Version [
        <sch:value-of select="$MinVersion"/>] or later is required. The latest version of
        <sch:value-of select="$SpecToCheck"/>is not being used in the validation infrastructure. Regardless of the version indicated on the instance document, the validation infrastructure needs
to use a version of
        <sch:value-of select="$SpecToCheck"/>that is version [
        <sch:value-of select="$MinVersion"/>] or later. NOTE: This is not an error of the instance document but of the validation environment itself. The incorrect value was found in
        <sch:value-of select="document-uri(document($pathToDocument))"/>
                    </sch:assert>
                </sch:rule>
            </sch:pattern>
```

3.2 - .//Lib/ValidateValidationEnvSchema.sch

Code Description

This abstract pattern checks to see if the validation environment has at least the version / revision of the Schema as of the writing of this specification. The calling rule must pass in $MinVersion, $SpecToCheck, $pathToDocument, $RuleID.

Schematron Code

```
<!--
    This abstract pattern checks to see the version of a Schema is greater than or equal to a passed in parameter.

    $MinVersion             := the version that SpecToCheck must be equal to or greater than.
    $SpecToCheck            := Name the spec whose version in the infrastructure is being checked.
    $pathToDocument         := Relative path to the document xsd that has ther version string
    $RuleID                 := The number of the rule in the concrete file.


-->

<sch:pattern xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                        abstract="true"
                        id="ValidateValidationEnvSchema">
            <sch:rule id="ValidateValidationEnvSchema-R1" context="/">
                <sch:assert test="document($pathToDocument)//xsd:schema//@version castable as xs:double and document($pathToDocument)//xsd:schema//@version &gt;= $MinVersion"
                        flag="error"
                        role="error">Version [
        <sch:value-of select="document($pathToDocument)//xsd:schema//@version"/>] of
        <sch:value-of select="$SpecToCheck"/>found; Version [
        <sch:value-of select="$MinVersion"/>] or later is required. The latest version of
        <sch:value-of select="$SpecToCheck"/>is not being used in the validation infrastructure. Regardless of the version indicated on the instance document, the validation infrastructure needs
to use a version of
        <sch:value-of select="$SpecToCheck"/>that is version [
        <sch:value-of select="$MinVersion"/>] or later. NOTE: This is not an error of the instance document but of the validation environment itself. The incorrect value was found in
        <sch:value-of select="document-uri(document($pathToDocument))"/>
                    </sch:assert>
                </sch:rule>
            </sch:pattern>
```

3.3 - .//Lib/ValidateValueExistenceInList.sch

Code Description

This abstract pattern checks to see if an attribute of an element exists in a list. The calling rule must pass $context, $searchTerm, $list, $errMsg.

Schematron Code

```xml
<!--
    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 xmlns:ism="urn:us:gov:ic:ism" is-a="ValidateValueExistenceInList" id="IRM_ID_00027" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:param name="context" value="//irm:CountryCode[@irm:vocabulary='FIPS']"/>
        <sch:param name="searchTerm" value="."/>
        <sch:param name="list" value="$coverageFipsDigraphList"/>
        <sch:param name="errMsg" value="'
            [IRM-ID-00027][Error]
            If element CountryCode has attribute vocabulary specified as FIPS
            the element value must be in CVEnumIRMCoverageFIPSDigraph.xml.
        '"/>
    </sch:pattern>

    Note: $iso4217TrigraphList is defined in the main document, IRM_XML.xml.
-->

<sch:pattern abstract="true" id="ValidateValueExistenceInList">
                <sch:rule id="ValidateValueExistenceInList-R1" context="$context">
                    <sch:assert test="some $token in $list satisfies $token = $searchTerm"
                                flag="error"
                                role="error">
                        <sch:value-of select="$errMsg"/>
                    </sch:assert>
                </sch:rule>
            </sch:pattern>
```

Chapter 4 - Schematron Schema

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

4.1 - .//IC-EDH_XML.sch

Code Description

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

Schematron Code

```xml
<!--UNCLASSIFIED-->
<?ICEA master?>
<!-- Notices - Distribution Notice:
        This document has been approved for Public Release and is available for use without restriction.
     -->
<!-- WARNING:
    Once compiled into an XSLT the result will
    be the aggregate classification of all the CVES
    and included .sch files
-->

<sch:schema queryBinding="xslt2">
                    <sch:ns uri="urn:us:gov:ic:pubs" prefix="pubs"/>
                    <sch:ns uri="urn:us:gov:ic:cve" prefix="cve"/>
                    <sch:ns uri="http://www.w3.org/1999/xlink" prefix="xlink"/>
                    <sch:ns uri="urn:us:gov:ic:ism" prefix="ism"/>
                    <sch:ns uri="urn:us:gov:ic:ntk" prefix="ntk"/>
                    <sch:ns uri="urn:us:gov:ic:edh" prefix="edh"/>
                    <sch:ns uri="urn:us:gov:ic:arh" prefix="arh"/>
                    <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:edh:xsl:util" prefix="util"/>
                    <sch:ns uri="http://www.w3.org/2001/XMLSchema" prefix="xs"/>
                    <sch:ns uri="http://www.functx.com" prefix="functx"/>
                    <!-- ************************************* -->
<!-- * Abstract Rule and Pattern Includes * -->
<!-- ************************************* -->

<sch:include href="./Lib/ValidateValueExistenceInList.sch"/>
                    <sch:include href="./Lib/ValidateValidationEnvCVE.sch"/>
                    <sch:include href="./Lib/ValidateValidationEnvSchema.sch"/>
                    <!--=====================-->
<!-- (U) Universal Lets -->
<!--=====================-->

<sch:let name="usagencyList"
                        value="document('../../CVE/USAgency/CVEnumUSAgencyAcronym.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
                    <!--*****************************-->
<!-- (U) Custom XSLT 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>
```

```xml
                        <xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                                      name="functx:next-day"
                                      as="xs:date?">
                    <xsl:param name="date" as="xs:anyAtomicType?"/>
                    <xsl:sequence select="xs:date($date) + xs:dayTimeDuration('P1D')"/>
                        </xsl:function>
                        <!--****************************-->
<!-- (U) IC-EDH Phases -->
<!--****************************-->
<!--****************************-->
<!-- (U) IC-EDH ID Rules -->
<!--****************************-->
<!--(U) -->

<sch:include href="./Rules/IC-EDH_ID_00001.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00002.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00003.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00004.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00005.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00006.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00008.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00011.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00012.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00013.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00014.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00015.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00016.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00017.sch"/>
                    <sch:include href="./Rules/IC-EDH_ID_00018.sch"/>
                        <!--****************************-->
<!-- (U) IC-EDH Phases -->
<!--****************************-->
</sch:schema>
                <!--UNCLASSIFIED-->
```

Chapter 5 - Removed Rules

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

5.1 - .//Rules/deleted/IC-EDH_ID_00007.sch

Rule Description

[IC-EDH-ID-00007][Error] Rule removed in V3. GUIDE format now enforced via the schema for the IC-ID specification. Human Readable: Rule removed in V3.

5.2 - .//Rules/deleted/IC-EDH_ID_00009.sch

Rule Description

[IC-EDH-ID-00009][Error] Rule removed in V2019-MAR due to ARH and NTK consolidation into ISM. Human Readable: Rule removed in V2019-MAR due to ARH and NTK consolidation into ISM.

5.3 - .//Rules/deleted/IC-EDH_ID_00010.sch

Rule Description

[IC-EDH-ID-00010][Error] Rule removed in V2019-MAR due to ARH and NTK consolidation into ISM. Human Readable: Rule removed in V2019-MAR due to ARH and NTK consolidation into ISM.