



Guide to Schematron Rules and Patterns

IC-TDF Schematron Guide

Version 2021-NOV

December 1, 2022

Distribution Notice:

This document has been approved for Public Release and is available for use without restriction.

Table of Contents

Chapter 1 - Introduction	1
1.1 - Purpose	1
1.2 - Overview	1
1.3 - Schematron	1
1.4 - Conformance	1
Chapter 2 - Rules	2
2.1 - ../Rules/IC-TDF_ID_00003.sch	3
2.2 - ../Rules/IC-TDF_ID_00004.sch	4
2.3 - ../Rules/IC-TDF_ID_00005.sch	5
2.4 - ../Rules/IC-TDF_ID_00016.sch	6
2.5 - ../Rules/IC-TDF_ID_00017.sch	7
2.6 - ../Rules/IC-TDF_ID_00018.sch	8
2.7 - ../Rules/IC-TDF_ID_00019.sch	9
2.8 - ../Rules/appliesToState/IC-TDF_ID_00026.sch	10
2.9 - ../Rules/appliesToState/IC-TDF_ID_00027.sch	11
2.10 - ../Rules/appliesToState/IC-TDF_ID_00028.sch	12
2.11 - ../Rules/appliesToState/IC-TDF_ID_00030.sch	13
2.12 - ../Rules/appliesToState/IC-TDF_ID_00031.sch	14
2.13 - ../Rules/IC-TDF_ID_00033.sch	15
2.14 - ../Rules/IC-TDF_ID_00034.sch	16
2.15 - ../Rules/IC-TDF_ID_00036.sch	17
2.16 - ../Rules/IC-TDF_ID_00042.sch	18
2.17 - ../Rules/IC-TDF_ID_00043.sch	19
2.18 - ../Rules/IC-TDF_ID_00044.sch	20
2.19 - ../Rules/IC-TDF_ID_00045.sch	21
2.20 - ../Rules/IC-TDF_ID_00046.sch	22
2.21 - ../Rules/IC-TDF_ID_00049.sch	23
2.22 - ../Rules/IC-TDF_ID_00050.sch	24
2.23 - ../Rules/IC-TDF_ID_00051.sch	25
2.24 - ../Rules/IC-TDF_ID_00052.sch	26
2.25 - ../Rules/IC-TDF_ID_00053.sch	27
2.26 - ../Rules/IC-TDF_ID_00054.sch	28
2.27 - ../Rules/IC-TDF_ID_00055.sch	29
2.28 - ../Rules/IC-TDF_ID_00056.sch	30
2.29 - ../Rules/IC-TDF_ID_00057.sch	31
2.30 - ../Rules/IC-TDF_ID_00058.sch	32
Chapter 3 - Abstract Patterns	33
3.1 - ../Lib/CompareVersionsInSkeleton.sch	34
3.2 - ../Lib/ValidateValidationEnvCVE.sch	35
3.3 - ../Lib/ValidateValidationEnvSchema.sch	36
Chapter 4 - Schematron Schema	37
4.1 - ../IC-TDF_XML.sch	38
Chapter 5 - Removed Rules	41
5.1 - ../Rules/deleted/IC-TDF_ID_00001.sch	41
5.2 - ../Rules/deleted/IC-TDF_ID_00002.sch	41
5.3 - ../Rules/deleted/IC-TDF_ID_00006.sch	41

5.4 - ../Rules/deleted/IC-TDF_ID_00007.sch	41
5.5 - ../Rules/deleted/IC-TDF_ID_00008.sch	41
5.6 - ../Rules/deleted/IC-TDF_ID_00009.sch	41
5.7 - ../Rules/deleted/IC-TDF_ID_00010.sch	41
5.8 - ../Rules/deleted/IC-TDF_ID_00011.sch	42
5.9 - ../Rules/deleted/IC-TDF_ID_00012.sch	42
5.10 - ../Rules/deleted/IC-TDF_ID_00013.sch	42
5.11 - ../Rules/deleted/IC-TDF_ID_00014.sch	42
5.12 - ../Rules/deleted/IC-TDF_ID_00015.sch	42
5.13 - ../Rules/deleted/IC-TDF_ID_00020.sch	42
5.14 - ../Rules/deleted/IC-TDF_ID_00021.sch	42
5.15 - ../Rules/deleted/IC-TDF_ID_00022.sch	43
5.16 - ../Rules/deleted/IC-TDF_ID_00023.sch	43
5.17 - ../Rules/deleted/IC-TDF_ID_00024.sch	43
5.18 - ../Rules/deleted/IC-TDF_ID_00025.sch	43
5.19 - ../Rules/deleted/IC-TDF_ID_00032.sch	43
5.20 - ../Rules/deleted/IC-TDF_ID_00035.sch	43
5.21 - ../Rules/deleted/IC-TDF_ID_00037.sch	43
5.22 - ../Rules/deleted/IC-TDF_ID_00038.sch	44
5.23 - ../Rules/deleted/IC-TDF_ID_00039.sch	44
5.24 - ../Rules/deleted/IC-TDF_ID_00040.sch	44
5.25 - ../Rules/deleted/IC-TDF_ID_00041.sch	44
5.26 - ../Rules/deleted/IC-TDF_ID_00047.sch	44
5.27 - ../Rules/deleted/IC-TDF_ID_00048.sch	44

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for IC-TDF. This guide is generated from the IC-TDF 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-TDF. 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-TDF. The abstract patterns may be used in numbered rules or provided as reference for use in rules developed by users of IC-TDF. 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-TDF 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-TDF 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-TDF 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-TDF's Schematron rules.

Chapter 2 - Rules

All of the numbered Rules for IC-TDF 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-TDF-ID-XXXXX, with rule files named IC-TDF_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-TDF_ID_00003.sch

Rule Description

[IC-TDF-ID-00003][Error] For element TrustedDataObject, there must be at least one element HandlingAssertion which specifies attribute scope containing [PAYL]. Human Readable: There must exist at least one handling marking for the payload.

Code Description

For each TrustedDataObject, this rule ensures that the count of HandlingAssertion element which specify attribute scope containing [PAYL] is greater than or equal to 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="IC-TDF-ID-00003">
  <sch:rule id="IC-TDF-ID-00003-R1" context="tdf:TrustedDataObject">
    <sch:assert test="count(child::tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('PAYL'))])>= 1"
      flag="error"
      role="error">[IC-TDF-ID-00003][Error] For element TrustedDataObject, there must be at least one element HandlingAssertion which specifies attribute scope
containing [PAYL]. Human Readable: There must exist at least one handling marking for the payload.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.2 - ../Rules/IC-TDF_ID_00004.sch

Rule Description

[IC-TDF-ID-00004][Error] For element TrustedDataObject, there must be exactly one element HandlingAssertion that specifies attribute scope containing [TDO] and contains an EDH element. Human Readable: There must exist a single EDH HandlingAssertion scoped for the entire TrustedDataObject.

Code Description

For element TrustedDataObject, this rule ensures that the count of HandlingAssertion elements that specify attribute scope containing [TDO] and have child::tdf:HandlingStatement/edh:Edh is exactly 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="IC-TDF-ID-00004">
    <sch:rule id="IC-TDF-ID-00004-R1" context="tdf:TrustedDataObject">
        <sch:assert test="count(child::tdf:HandlingAssertion[child::tdf:HandlingStatement/edh:Edh and @tdf:scope = 'TDO'])= 1"
            flag="error"
            role="error">[IC-TDF-ID-00004][Error] For element TrustedDataObject, there must be exactly one element HandlingAssertion that specifies attribute scope
containing [TDO] and contains an EDH element.</sch:assert>
        </sch:rule>
    </sch:pattern>
```


2.3 - ../Rules/IC-TDF_ID_00005.sch

Rule Description

[IC-TDF-ID-00005][Error] For element TrustedDataCollection, there must be exactly one element HandlingAssertion that specifies @scope="TDC" and contains an EDH element. Human Readable: There must exist a single EDH HandlingAssertion scoped for the entire TrustedDataCollection.

Code Description

For element TrustedDataCollection, this rule ensures that the count of HandlingAssertion elements that specify attribute scope containing [TDC] and contain an EDH element is exactly 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="IC-TDF-ID-00005">
  <sch:rule id="IC-TDF-ID-00005-R1" context="tdf:TrustedDataCollection">
    <sch:assert test="count(child::tdf:HandlingAssertion[child::tdf:HandlingStatement/edh:Edh and @tdf:scope = 'TDC'])= 1"
      flag="error"
      role="error">[IC-TDF-ID-00005][Error] For element TrustedDataCollection, there must be exactly one element HandlingAssertion that specifies @scope="TDC" and
contains an EDH element. Human Readable: There must exist a single EDH HandlingAssertion scoped for the entire TrustedDataCollection.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.4 - ../Rules/IC-TDF_ID_00016.sch

Rule Description

[IC-TDF-ID-00016][Error] EDH HandlingAssertions with TDO scope must have an ARH security element has ism:resourceElement="true". Human Readable: An EDH HandlingAssertion with scope pertaining to the entire TrustedDataObject (TDO) must declare itself a resource level object.

Code Description

EDH HandlingAssertions with scope containing [TDO], ensure that its descendant ARH element, Security or ExternalSecurity, has ism:resourceElement="true".

Schematron Code

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

<sch:pattern id="IC-TDF-ID-00016">
    <sch:rule id="IC-TDF-ID-00016-R1"
        context="tdf:HandlingAssertion[child::tdf:HandlingStatement/edh:Edh and util:containsAnyOfTheTokens(@tdf:scope, ('TDO'))]">
        <sch:assert test="descendant::arh:*[@ism:resourceElement=true()]"
            flag="error"
            role="error">[IC-TDF-ID-00016][Error] HandlingAssertions with scope containing the token [TDO] must have an EDH whose ARH security element has
ism:resourceElement="true" specified. Human Readable: An EDH HandlingAssertion with scope pertaining to the entire TrustedDataObject (TDO) must declare itself a resource level object.</
sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.5 - ../Rules/IC-TDF_ID_00017.sch

Rule Description

[IC-TDF-ID-00017][Error] EDH HandlingAssertions with scope containing the token [TDC] must have an EDH whose ARH security element has ism:resourceElement="true" specified. Human Readable: When a HandlingAssertion has scope pertaining to the entire TrustedDataCollection (TDC) it must declare itself a resource level object.

Code Description

Where an EDH HandlingAssertion exists with scope containing [TDC], this rule ensures that its descendant ARH element, Security or ExternalSecurity, has ism:resourceElement specified with a value of true.

Schematron Code

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

<sch:pattern id="IC-TDF-ID-00017">
    <sch:rule id="IC-TDF-ID-00017-R1"
        context="tdf:HandlingAssertion[child::tdf:HandlingStatement/edh:Edh and util:containsAnyOfTheTokens(@tdf:scope, ('TDC'))]">
        <sch:assert test="descendant::arh:*[@ism:resourceElement=true()]"
            flag="error"
            role="error">[IC-TDF-ID-00017][Error] HandlingAssertions with scope containing the token [TDC] must have an EDH whose ARH security element has
ism:resourceElement="true" specified. Human Readable: When a HandlingAssertion has scope pertaining to the entire TrustedDataCollection (TDC) it must declare itself a resource level object.</
sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.6 - ../Rules/IC-TDF_ID_00018.sch

Rule Description

[IC-TDF-ID-00018][Error] HandlingAssertions with scope containing the token [TDO] cannot use the ExternalEdh child element. Human Readable: When a HandlingAssertion has scope pertaining to the entire TrustedDataObject (TDO), it must never use the ExternalEdh child element because the HandlingAssertion will always refer to the object in which it resides.

Code Description

Where a HandlingAssertion exists with scope containing [TDO], this rule ensures that it does not have a child of ExternalEdh.

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-TDF-ID-00018">
    <sch:rule id="IC-TDF-ID-00018-R1"
        context="tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('TDO'))]">
        <sch:assert test="not(descendant::edh:ExternalEdh)" flag="error" role="error">[IC-TDF-ID-00018][Error] HandlingAssertions with scope containing the token [TDO] cannot
use the ExternalEdh child element. Human Readable: When a HandlingAssertion has scope pertaining to the entire TrustedDataObject (TDO), it must never use the ExternalEdh child element because
the HandlingAssertion will always refer to the object in which it resides.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.7 - ../Rules/IC-TDF_ID_00019.sch

Rule Description

[IC-TDF-ID-00019][Error] HandlingAssertions with scope containing the token [TDC] cannot use the ExternalEdh child element. Human Readable: When a HandlingAssertion has scope pertaining to the entire TrustedDataCollection (TDC), it must never use the ExternalEdh child element because the HandlingAssertion will always refer to the Collection in which it resides.

Code Description

Where a HandlingAssertion exists with scope containing [TDC], this rule ensures that it does not have a child of ExternalEdh.

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-TDF-ID-00019">
    <sch:rule id="IC-TDF-ID-00019-R1"
        context="tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('TDC'))]">
        <sch:assert test="not(descendant::edh:ExternalEdh)" flag="error" role="error">[IC-TDF-ID-00019][Error] HandlingAssertions with scope containing the token [TDC] cannot
use the ExternalEdh child element. Human Readable: When a HandlingAssertion has scope pertaining to the entire TrustedDataCollection (TDC), it must never use the ExternalEdh child element
because the HandlingAssertion will always refer to the Collection in which it resides.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.8 - ../Rules/appliesToState/IC-TDF_ID_00026.sch

Rule Description

[IC-TDF-ID-00026][Error] If payload attribute @isEncrypted="true", then there needs to be two handling assertions with attribute scope="PAYL": one with attribute @appliesToState="encrypted" and the other with attribute appliesToState="unencrypted". Human Readable: Encrypted payloads require handling assertions for both encrypted and unencrypted payload states.

Code Description

If there exists a TDO payload element with attribute @isEncrypted as true, this rule ensures there is one handling assertion of @scope PAYL and @appliestostate of encrypted, and one handling assertion of @scope PAYL and @appliestostate of unencrypted.

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-TDF-ID-00026">
    <sch:rule id="IC-TDF-ID-00026-R1"
        context="tdf:TrustedDataObject/tdf:*[@tdf:isEncrypted=true()]">
        <sch:assert test="count(parent::node()/tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('PAYL')) and @tdf:appliesToState='encrypted'])= 1 and
count(parent::node()/tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('PAYL')) and @tdf:appliesToState='unencrypted'])= 1"
            flag="error"
            role="error">[IC-TDF-ID-00026][Error] If payload attribute @isEncrypted="true", then there needs to be two handling assertions with attribute scope="PAYL":
one with attribute @appliesToState="encrypted" and the other with attribute appliesToState="unencrypted".</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.9 - ../Rules/appliesToState/IC-TDF_ID_00027.sch

Rule Description

[IC-TDF-ID-00027][Error] If payload attribute @isEncrypted="true", the handling assertion with @scope="PAYL" that contains @appliesToState="unencrypted" must contain an edh:externalEDH. Human Readable: When content is encrypted, the handling assertion describing the content in an unencrypted state is in effect external.

Code Description

If there exists a TDO payload element with attribute @isEncrypted as true, this rule ensures that there is one handling assertion of @scope PAYL, @appliestostate of unencrypted, and has descendant element ExternalEdh.

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-TDF-ID-00027">
    <sch:rule id="IC-TDF-ID-00027-R1"
        context="tdf:TrustedDataObject/tdf:*[@tdf:isEncrypted=true()]">
        <sch:assert test="count(parent::node()/tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('PAYL')) and @tdf:appliesToState='unencrypted']/
tdf:HandlingStatement/edh:ExternalEdh)= 1"
            flag="error"
            role="error">[IC-TDF-ID-00027][Error] If payload attribute @isEncrypted="true", the handling assertion with @scope="PAYL" that contains
@appliesToState="unencrypted" must contain an edh:externalEDH. Human Readable: When content is encrypted, the handling assertion describing the content in an unencrypted state is in effect
external.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.10 - ./Rules/appliesToState/IC-TDF_ID_00028.sch

Rule Description

[IC-TDF-ID-00028][Error] If payload attribute @isEncrypted="true" and the payload is not external, the handling assertion with @scope="PAYL" that contains @appliesToState="encrypted" must contain a regular edh:EDH. Human Readable: Internal content requires an EDH.

Code Description

Given a TDO with an internal payload with attribute @isEncrypted="true", the handling assertion with @scope="PAYL" that contains @appliesToState="encrypted" must contain a regular edh:EDH.

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-TDF-ID-00028">
    <sch:rule id="IC-TDF-ID-00028-R1"
        context="tdf:TrustedDataObject[tdf:StringPayload/@tdf:isEncrypted=true()] | tdf:TrustedDataObject[tdf:Base64BinaryPayload/@tdf:isEncrypted=true()] |
tdf:TrustedDataObject[tdf:StructuredPayload/@tdf:isEncrypted=true()]">
        <sch:assert test="count(tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope, ('PAYL')) and @tdf:appliesToState='encrypted']/tdf:HandlingStatement/edh:Edh)= 1"
            flag="error"
            role="error">[IC-TDF-ID-00028][Error] If payload attribute @isEncrypted="true" and the payload is not external, the handling assertion with @scope="PAYL"
that contains @appliesToState="encrypted" must contain a regular edh:EDH. Human Readable: Internal content requires an EDH.</sch:assert>
        </sch:rule>
    </sch:pattern>
```


2.11 - ./Rules/appliesToState/IC-TDF_ID_00030.sch

Rule Description

[IC-TDF-ID-00030][Error] If statement attribute @isEncrypted="true", the statement metadata that contains @appliesToState="unencrypted" must contain an arh:ExternalSecurity Human Readable: When statement content is encrypted, the handling statement describing the content in an unencrypted state is in effect external.

Code Description

Given a TDO with an encrypted assertion (statement attribute @isEncrypted="true"), the statement metadata that contains @appliesToState="unencrypted" must contain an arh:ExternalSecurity.

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-TDF-ID-00030">
  <sch:rule id="IC-TDF-ID-00030-R1"
    context="tdf:TrustedDataObject/tdf:Assertion/tdf:*[@tdf:isEncrypted=true()]">
    <sch:assert test="count(parent::node()/tdf:StatementMetadata[@tdf:appliesToState='unencrypted' and descendant-or-self::arh:ExternalSecurity])= 1"
      flag="error"
      role="error">[IC-TDF-ID-00030][Error] If statement attribute @isEncrypted="true", the statement metadata that contains @appliesToState="unencrypted" must
contain an arh:ExternalSecurity Human Readable: When statement content is encrypted, the handling statement describing the content in an unencrypted state is in effect external.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.12 - ../Rules/appliesToState/IC-TDF_ID_00031.sch

Rule Description

[IC-TDF-ID-00031][Error] If assertion statement attribute @isEncrypted="true", then there needs to be two statement metadata elements: one with attribute @appliesToState="encrypted" and the other with attribute appliesToState="unencrypted". Human Readable: If an assertion statement is encrypted, it must have statement metadata to describe handling for both its encrypted state, and unencrypted state.

Code Description

If a TDO has an encrypted assertion (@isEncrypted="true"), then there needs to be two statement metadata elements: one with attribute @appliesToState="encrypted" and the other with attribute appliesToState="unencrypted".

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-TDF-ID-00031">
    <sch:rule id="IC-TDF-ID-00031-R1"
        context="tdf:TrustedDataObject/tdf:Assertion/tdf:*[@tdf:isEncrypted=true()]">
        <sch:assert test="count(parent::node()/tdf:StatementMetadata[@tdf:appliesToState='encrypted'])= 1 and count(parent::node()/tdf:StatementMetadata[@tdf:appliesToState='unencrypted'])= 1"
            flag="error"
            role="error">[IC-TDF-ID-00031][Error] If assertion statement attribute @isEncrypted="true", then there needs to be two statement metadata elements: one with attribute @appliesToState="encrypted" and the other with attribute appliesToState="unencrypted". Human Readable: If an assertion statement is encrypted, it must have statement metadata to describe handling for both for its encrypted state, and unencrypted state.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.13 - ../Rules/IC-TDF_ID_00033.sch

Rule Description

[IC-TDF-ID-00033][Error] A TrustedDataObject with a ReferencePayload must have an ExternalEDH element in the HandlingAssertion with scope [PAYL].

Code Description

For TrustedDataObject elements with a ReferencePayload, ensure that the HandlingAssertion with scope [PAYL] has an ExternalEDH 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-TDF-ID-00033">
    <sch:rule id="IC-TDF-ID-00033-R1"
        context="tdf:TrustedDataObject//tdf:ReferenceValuePayload">
        <sch:assert test="ancestor::tdf:TrustedDataObject//tdf:HandlingAssertion[@tdf:scope='PAYL']//edh:ExternalEdh"
            flag="error"
            role="error">[IC-TDF-ID-00033][Error] A TrustedDataObject with a ReferencePayload must have an ExternalEDH element in the HandlingAssertion with scope
[PAYL].</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.14 - ../Rules/IC-TDF_ID_00034.sch

Rule Description

[IC-TDF-ID-00034][Error] An Assertion with a ReferenceStatement must have an ExternalEDH or ExternalSecurity element in the StatementMetadata.

Code Description

For Assertion elements with a ReferenceStatement, ensure that the StatementMetadata has an ExternalEDH or ExternalSecurity 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-TDF-ID-00034">
    <sch:rule id="IC-TDF-ID-00034-R1" context="tdf:Assertion//tdf:ReferenceStatement">
        <sch:assert test="ancestor::tdf:Assertion/tdf:StatementMetadata[edh:ExternalEdh or arh:ExternalSecurity]"
                    flag="error"
                    role="error">[IC-TDF-ID-00034][Error] An Assertion with a ReferenceStatement must have an ExternalEDH or ExternalSecurity element in the StatementMetadata.</
sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.15 - ../Rules/IC-TDF_ID_00036.sch

Rule Description

[IC-TDF-ID-00036][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'IC-EDH' 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-TDF-ID-00036" is-a="ValidateValidationEnvSchema">
    <sch:param name="MinVersion" value="'201903'"/>
    <sch:param name="SpecToCheck" value="'IC-EDH'"/>
    <sch:param name="pathToDocument" value="'../..Schema/IC-EDH/IC-EDH.xsd'"/>
    <sch:param name="RuleID" value="'IC-TDF-ID-00036'"/>
</sch:pattern>
```

2.16 - ../Rules/IC-TDF_ID_00042.sch

Rule Description

[IC-TDF-ID-00042][Error] The first HandlingAssertion of a TDF must have the attribute scope with a value of [TDO] or [TDC] and contain an EDH.

Code Description

This rule triggers on the first HandlingAssertion element for each TDF and tests that the value of the @tdf:scope attribute is set a value of [TDO] or [TDC] and that an EDH exists. Otherwise, an error is triggered. This prevents some other handling assertion such as Revision Recall from being the ISM resource node for the entire TDO.

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-TDF-ID-00042">
    <sch:rule id="IC-TDF-ID-00042-R1" context="tdf:*/tdf:HandlingAssertion[1]">
        <sch:assert test="((parent::tdf:TrustedDataObject and @tdf:scope='TDO') or (parent::tdf:TrustedDataCollection and @tdf:scope='TDC')) and ./tdf:HandlingStatement/edh:Edh"
            flag="error"
            role="error">[IC-TDF-ID-00042][Error] The first HandlingAssertion of a TDF must have the attribute scope with a value of [TDO] or [TDC] and contain an EDH.</
sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.17 - ../Rules/IC-TDF_ID_00043.sch

Rule Description

[IC-TDF-ID-00043][Error] ntk:Access elements on portions of a TDO must be rolled up to the resource level. As such there must be an ntk:Access on the HandlingAssertion with scope [TDO]. Precise rollup is left to the creator to determine.
Human Readable: If there is an ntk:Access in any portion of a TDO, then there must be an ntk:Access in the HandlingAssertion with scope="TDO"

Code Description

This rule triggers on any ntk:Access that exists except for one in the tdf:HandlingAssertion with a scope [TDO]. If it triggers it checks that there is an ntk:Access in the HandlingAssertion with scope [TDO] otherwise it sets an error.

Schematron Code

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

<sch:pattern id="IC-TDF-ID-00043">
    <sch:rule id="IC-TDF-ID-00043-R1"
        context="tdf:TrustedDataObject//ntk:Access[not(ancestor::tdf:HandlingAssertion[@tdf:scope='TDO'] or @ntk:externalReference=true())]">
        <sch:assert test="ancestor::tdf:TrustedDataObject/tdf:HandlingAssertion[@tdf:scope='TDO']//ntk:Access"
            flag="error"
            role="error">[IC-TDF-ID-00043][Error] If there is an ntk:Access in any portion of a TDO, then there must be an ntk:Access in the HandlingAssertion with
scope="TDO"</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.18 - ../Rules/IC-TDF_ID_00044.sch

Rule Description

[IC-TDF-ID-00044][Error] ntk:Access elements on child TDOs or Assertions must be rolled up to the resource level. As such there must be an ntk:Access on the HandlingAssertion with scope [TDC]. Precise rollup is left to the creator to determine. Human Readable: If there is an ntk:Access in any portion of a TDC, then there must be an ntk:Access in the HandlingAssertion with scope="TDC"

Code Description

This rule triggers on any ntk:Access that exists except for one in the tdf:HandlingAssertion with a scope [TDC]. If it triggers it checks that there is an ntk:Access in the HandlingAssertion with scope [TDC] otherwise it sets an error.

Schematron Code

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

<sch:pattern id="IC-TDF-ID-00044">
    <sch:rule id="IC-TDF-ID-00044-R1"
        context="tdf:TrustedDataCollection//ntk:Access[not(ancestor::tdf:HandlingAssertion[@tdf:scope='TDC'] or @ntk:externalReference=true())]">
        <sch:assert test="ancestor::tdf:TrustedDataCollection/tdf:HandlingAssertion[@tdf:scope='TDC']//ntk:Access"
            flag="error"
            role="error">[IC-TDF-ID-00044][Error] If there is an ntk:Access in any portion of a TDC, then there must be an ntk:Access in the HandlingAssertion with
scope="TDC"</sch:assert>
        </sch:rule>
    </sch:pattern>
```


2.19 - ../Rules/IC-TDF_ID_00045.sch

Rule Description

[IC-TDF-ID-00045][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'RevRecall' that is version '202111' (Version:2021-NOV) 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-TDF-ID-00045" is-a="ValidateValidationEnvSchema">
    <sch:param name="MinVersion" value="'202111'"/>
    <sch:param name="SpecToCheck" value="'RevRecall'"/>
    <sch:param name="pathToDocument"
        value="'../../Schema/RevRecall/RevRecall_XML.xsd'"/>
    <sch:param name="RuleID" value="'IC-TDF-ID-00045'"/>
</sch:pattern>
```

2.20 - .//Rules/IC-TDF_ID_00046.sch

Rule Description

[IC-TDF-ID-00046][Error] If there exist multiple version attributes in the same namespace within the TDF skeleton, then they must specify the same version number. Human Readable: The ism:DESVersion declared for a specification must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions.

Code Description

This rule uses an abstract pattern to consolidate logic. For all ism:DESVersion attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendents of tdf:HandlingAssertion or tdf:StatementMetadata elements.

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-TDF-ID-00046" is-a="CompareVersionsInSkeleton">
    <sch:param name="context"
        value="//tdf:*[@ism:DESVersion] | //tdf:HandlingAssertion/*[@ism:DESVersion] | //tdf:StatementMetdata/*[@ism:DESVersion]"/>
    <sch:param name="namespace" value="'urn:us:gov:ic:ism'"/>
    <sch:param name="attrName" value="'DESVersion'"/>
    <sch:param name="ruleID" value="'IC-TDF-ID-00046'"/>
</sch:pattern>
```

2.21 - ./Rules/IC-TDF_ID_00049.sch

Rule Description

[IC-TDF-ID-00049][Error] If there exist multiple version attributes in the same namespace within the TDF skeleton, then they must specify the same version number. Human Readable: The edh:DESVersion declared for a specification must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions.

Code Description

This rule uses an abstract pattern to consolidate logic. For all edh:DESVersion attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendents of tdf:HandlingAssertion or tdf:StatementMetadata elements.

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-TDF-ID-00049" is-a="CompareVersionsInSkeleton">
    <sch:param name="context"
        value="tdf:*[@edh:DESVersion] | tdf:HandlingAssertion/*[@edh:DESVersion] | tdf:StatementMetdata/*[@edh:DESVersion]"/>
    <sch:param name="namespace" value="'urn:us:gov:ic:edh'"/>
    <sch:param name="attrName" value="'DESVersion'"/>
    <sch:param name="ruleID" value="'IC-TDF-ID-00049'"/>
</sch:pattern>
```

2.22 - .//Rules/IC-TDF_ID_00050.sch

Rule Description

[IC-TDF-ID-00050][Error] If there exist multiple version attributes in the same namespace within the TDF skeleton, then they must specify the same version number. Human Readable: The icid:DESVersion declared for a specification must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions.

Code Description

This rule uses an abstract pattern to consolidate logic. For all icid:DESVersion attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendents of tdf:HandlingAssertion or tdf:StatementMetadata elements.

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-TDF-ID-00050" is-a="CompareVersionsInSkeleton">
    <sch:param name="context"
        value="tdf:*[@icid:DESVersion] | tdf:HandlingAssertion/*[@icid:DESVersion] | tdf:StatementMetdata/*[@icid:DESVersion]"/>
    <sch:param name="namespace" value="'urn:us:gov:ic:id'"/>
    <sch:param name="attrName" value="'DESVersion'"/>
    <sch:param name="ruleID" value="'IC-TDF-ID-00050'"/>
</sch:pattern>
```

2.23 - ../Rules/IC-TDF_ID_00051.sch

Rule Description

[IC-TDF-ID-00051][Error] If there exist multiple version attributes in the same namespace within the TDF skeleton, then they must specify the same version number. Human Readable: The usagency:CESVersion declared for a specification must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions.

Code Description

This rule uses an abstract pattern to consolidate logic. For all usagency:CESVersion attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendents of tdf:HandlingAssertion or tdf:StatementMetadata elements.

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-TDF-ID-00051" is-a="CompareVersionsInSkeleton">
    <sch:param name="context"
        value="tdf:*[@usagency:CESVersion] | tdf:HandlingAssertion//*[@usagency:CESVersion] | tdf:StatementMetdata//*[@usagency:CESVersion]"/>
    <sch:param name="namespace" value="'urn:us:gov:ic:usagency'"/>
    <sch:param name="attrName" value="'CESVersion'"/>
    <sch:param name="ruleID" value="'IC-TDF-ID-00051'"/>
</sch:pattern>
```

2.24 - ./Rules/IC-TDF_ID_00052.sch

Rule Description

[IC-TDF-ID-00052][Error] If there exist multiple version attributes in the same namespace within the TDF skeleton, then they must specify the same version number. Human Readable: The tdf:version declared for a specification must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions.

Code Description

This rule uses an abstract pattern to consolidate logic. For all tdf:version attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendents of tdf:HandlingAssertion or tdf:StatementMetadata elements.

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-TDF-ID-00052" is-a="CompareVersionsInSkeleton">
    <sch:param name="context"
        value="tdf:*[@tdf:version] | tdf:HandlingAssertion/*[@tdf:version] | tdf:StatementMetdata/*[@tdf:version]"/>
    <sch:param name="namespace" value="'urn:us:gov:ic:tdf'"/>
    <sch:param name="attrName" value="'version'"/>
    <sch:param name="ruleID" value="'IC-TDF-ID-00052'"/>
</sch:pattern>
```

2.25 - .//Rules/IC-TDF_ID_00053.sch

Rule Description

[IC-TDF-ID-00053][Error] If there exist multiple version attributes in the same namespace within the TDF skeleton, then they must specify the same version number. Human Readable: The ism:ISMCATCESVersion declared for a specification must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions.

Code Description

This rule uses an abstract pattern to consolidate logic. For all ism:ISMCATCESVersion attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendents of tdf:HandlingAssertion or tdf:StatementMetadata elements.

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-TDF-ID-00053" is-a="CompareVersionsInSkeleton">
    <sch:param name="context"
        value="//tdf:*[@ism:ISMCATCESVersion] | //tdf:HandlingAssertion/*[@ism:ISMCATCESVersion] | //tdf:StatementMetdata/*[@ism:ISMCATCESVersion]"/>
    <sch:param name="namespace" value="'urn:us:gov:ic:ism'"/>
    <sch:param name="attrName" value="'ISMCATCESVersion'"/>
    <sch:param name="ruleID" value="'IC-TDF-ID-00053'"/>
</sch:pattern>
```

2.26 - ./Rules/IC-TDF_ID_00054.sch

Rule Description

[IC-TDF-ID-00054][Warning] tdf:version attribute SHOULD be specified as version 202111-IC-TDF.202111 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 "^202111-IC-TDF.202111(-.{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-TDF-ID-00054">
  <sch:rule id="IC-TDF-ID-00054-R1" context="*[@tdf:version]">
    <sch:assert test="matches(@tdf:version, '^202111-IC-TDF.202111(-.{1,23})?$')"
      flag="warning"
      role="warning">[IC-TDF-ID-00054][Warning] tdf:version attribute SHOULD be specified as version 202111-IC-TDF.202111 with an optional extension. Found:
  <sch:value-of select="@tdf:version"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```


2.27 - ../Rules/IC-TDF_ID_00055.sch

Rule Description

[IC-TDF-ID-00055][Error] For element TrustedDataObject whose payload is NOT encrypted, there must not be more than one element HandlingAssertion that specifies attribute scope containing [PAYL] and contains an EDH element. Human Readable: For TrustedDataObjects with unencrypted payloads, there must not be more than a single EDH HandlingAssertion scoped for the payload.

Code Description

For element TrustedDataObject whose payload is NOT encrypted, ensure that the count of HandlingAssertion elements that specify attribute scope containing [PAYL] and have child::tdf:HandlingStatement/edh:Edh is not more than 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="IC-TDF-ID-00055">
  <sch:rule id="IC-TDF-ID-00055-R1"
    context="tdf:TrustedDataObject[not(tdf:*/@tdf:isEncrypted=true())]">
    <sch:assert test="not(count(child::tdf:HandlingAssertion[child::tdf:HandlingStatement/edh:Edh and @tdf:scope = 'PAYL']) &gt; 1)"
      flag="error"
      role="error">[IC-TDF-ID-00055][Error] For element TrustedDataObject whose payload is NOT encrypted, there must not be more than one element
HandlingAssertion that specifies attribute scope containing [PAYL] and contains an EDH element. Human Readable: For TrustedDataObjects with unencrypted payloads, there must not be more than a
single EDH HandlingAssertion scoped for the payload.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.28 - ../Rules/IC-TDF_ID_00056.sch

Rule Description

[IC-TDF-ID-00056][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'ISM' that is version '202111' (Version:2021-NOV) 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-TDF-ID-00056" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'202111'"/>
    <sch:param name="SpecToCheck" value="'ISM'"/>
    <sch:param name="pathToDocument"
        value="'../../../../CVE/ISM/CVEnumISMClassificationAll.xml'"/>
    <sch:param name="RuleID" value="'IC-TDF-ID-00056'"/>
</sch:pattern>
```

2.29 - ../Rules/IC-TDF_ID_00057.sch

Rule Description

[IC-TDF-ID-00057][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'IC-SF' that is version '202111' (Version:2021-NOV) 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-TDF-ID-00057" is-a="ValidateValidationEnvSchema">
    <sch:param name="MinVersion" value="'202111'"/>
    <sch:param name="SpecToCheck" value="'IC-SF'"/>
    <sch:param name="pathToDocument" value="'../Schema/IC-SF/IC-SF.xsd'"/>
    <sch:param name="RuleID" value="'IC-TDF-ID-00057'"/>
</sch:pattern>
```

2.30 - ../Rules/IC-TDF_ID_00058.sch

Rule Description

[IC-TDF-ID-00058][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'BASE-TDF' that is version '202111' (Version:2021-NOV) 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-TDF-ID-00058" is-a="ValidateValidationEnvSchema">
    <sch:param name="MinVersion" value="'202111'"/>
    <sch:param name="SpecToCheck" value="'BASE-TDF'"/>
    <sch:param name="pathToDocument" value="'../Schema/BASE-TDF/BASE-TDF.xsd'"/>
    <sch:param name="RuleID" value="'IC-TDF-ID-00058'"/>
</sch:pattern>
```

Chapter 3 - Abstract Patterns

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

3.1 - ./Lib/CompareVersionsInSkeleton.sch

Code Description

For all ism:DESVersion attributes found on the TDF skeleton, ensure all the versions are the same. The TDF Skeleton includes the TDF elements themselves and descendants of tdf:HandlingAssertion or tdf:StatementMetadata elements.

Schematron Code

```
<!--
  This abstract pattern checks all ism:DESVersion attributes found on the TDF skeleton to
  ensure that all the versions are the same. The TDF Skeleton includes the TDF elements
  themselves and descendants of tdf:HandlingAssertion or tdf:StatementMetadata elements.

  List of params in the form "paramName : example of what to do"
  context      := the nodes that the rule will test.
                  //tdf:*[@ism:DESVersion] | //tdf:HandlingAssertion/*[@ism:DESVersion]
                  | //tdf:StatementMetdata/*[@ism:DESVersion]
  namespace    := the namespace of the attribute to be tested i.e. urn:us:gov:ic:ism
  attrName     := the attribute that will be tested i.e. DESVersion
  ruleID       := the rule name that invokes the abstrat rule, for use in error message
                  i.e. IC-TDF-ID-00046
-->

<sch:pattern abstract="true" id="CompareVersionsInSkeleton">
  <sch:rule id="CompareVersionsInSkeleton-R1" context="$context">
    <sch:assert test="every $ver in (//tdf:*/*[@local-name()=$attrName and namespace-uri()=$namespace] | //tdf:HandlingAssertion/*[@local-name()=$attrName and namespace-
uri()=$namespace] | //tdf:StatementMetadata/*[@local-name()=$attrName and namespace-uri()=$namespace]) satisfies (number(if (contains($ver,'-')) then substring-before($ver,'-') else
$ver))=(number(if (contains(@*[local-name()=$attrName and namespace-uri()=$namespace],'-')) then substring-before(@*[local-name()=$attrName and namespace-uri()=$namespace],'-') else @*[local-
name()=$attrName and namespace-uri()=$namespace]))"
      flag="error"
      role="error">[
    <sch:value-of select="$ruleID"/>][Error] The {
    <sch:value-of select="$namespace"/>}
    <sch:value-of select="$attrName"/>declared must be the same throughout the IC-TDF skeleton including the HandlingAssertions and StatementMetadata within assertions. Versions found:
    <sch:value-of select="string-join(distinct-values(for $ver in (//tdf:*/*[@local-name()=$attrName and namespace-uri()=$namespace] | //tdf:HandlingAssertion/*[@local-name()=$attrName and
namespace-uri()=$namespace] | //tdf:StatementMetadata/*[@local-name()=$attrName and namespace-uri()=$namespace]) return $ver), ', ')" />
      </sch:assert>
    </sch:rule>
  </sch:pattern>
```

3.2 - ./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

```
<!--
  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 >= $MinVersion"
               flag="error"
               role="error">[
  <sch:value-of select="$RuleID"/>][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>
```

3.3 - ./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 >= $MinVersion"
               flag="error"
               role="error">[
  <sch:value-of select="$RuleID"/>][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>
```


Chapter 4 - Schematron Schema

The top level Schematron file for IC-TDF 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-TDF_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

```
<!--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 prefix="xs" uri="http://www.w3.org/2001/XMLSchema"/>
    <sch:ns prefix="cve" uri="urn:us:gov:ic:cve"/>
    <sch:ns prefix="cve" uri="urn:us:gov:ic:cve"/>
    <sch:ns prefix="tdf" uri="urn:us:gov:ic:tdf"/>
    <sch:ns prefix="ism" uri="urn:us:gov:ic:ism"/>
    <sch:ns prefix="arh" uri="urn:us:gov:ic:arh"/>
    <sch:ns prefix="edh" uri="urn:us:gov:ic:edh"/>
    <sch:ns prefix="ntk" uri="urn:us:gov:ic:ntk"/>
    <sch:ns prefix="icid" uri="urn:us:gov:ic:id"/>
    <sch:ns prefix="usagency" uri="urn:us:gov:ic:usagency"/>
    <sch:ns prefix="revrecall" uri="urn:us:gov:ic:revrecall"/>
    <sch:ns prefix="util" uri="urn:us:gov:ic:tdf:xsl:util"/>
    <!-- (U) Abstract Patterns -->

<sch:include href="./Lib/CompareVersionsInSkeleton.sch"/>
    <sch:include href="./Lib/ValidateValidationEnvCVE.sch"/>
    <sch:include href="./Lib/ValidateValidationEnvSchema.sch"/>
    <!--*****-->
<!-- (U) Utility functions -->
<!--*****-->
<!--
    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 true if every token in the attribute is contained in the provided list.
-->

<xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    name="util:containsOnlyTheTokens"
    as="xs:boolean">
    <xsl:param name="attribute"/>
```

```

        <xsl:param name="tokenList" as="xs:string+"/>
        <xsl:value-of select="every $attrToken in tokenize(normalize-space(string($attribute)), ' ') satisfies $attrToken = $tokenList"/>
    </xsl:function>
    <!--*****-->
<!-- (U) IC-TDF Phases -->
<!--*****-->
<!--*****-->
<!-- (U) IC-TDF ID Rules -->
<!--*****-->
<!--(U) appliesToState-->

<sch:include href="./Rules/appliesToState/IC-TDF_ID_00026.sch"/>
    <sch:include href="./Rules/appliesToState/IC-TDF_ID_00027.sch"/>
    <sch:include href="./Rules/appliesToState/IC-TDF_ID_00028.sch"/>
    <sch:include href="./Rules/appliesToState/IC-TDF_ID_00030.sch"/>
    <sch:include href="./Rules/appliesToState/IC-TDF_ID_00031.sch"/>
    <!--(U) -->

<sch:include href="./Rules/IC-TDF_ID_00003.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00004.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00005.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00016.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00017.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00018.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00019.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00033.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00034.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00036.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00042.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00043.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00044.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00045.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00046.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00049.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00050.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00051.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00052.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00053.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00054.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00055.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00056.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00057.sch"/>
    <sch:include href="./Rules/IC-TDF_ID_00058.sch"/>
    <!--*****-->
<!-- (U) IC-TDF Phases -->
<!--*****-->
</sch:schema>

    <!-- UNCLASSIFIED -->
<!--UNCLASSIFIED-->
```

Chapter 5 - Removed Rules

All of the numbered Rules for IC-TDF 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-TDF_ID_00001.sch`

Rule Description

[IC-TDF-ID-00001][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00005.

5.2 - `./Rules/deleted/IC-TDF_ID_00002.sch`

Rule Description

[IC-TDF-ID-00002][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00006.

5.3 - `./Rules/deleted/IC-TDF_ID_00006.sch`

Rule Description

[IC-TDF-ID-00006][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00007.

5.4 - `./Rules/deleted/IC-TDF_ID_00007.sch`

Rule Description

[IC-TDF-ID-00007][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00008.

5.5 - `./Rules/deleted/IC-TDF_ID_00008.sch`

Rule Description

[IC-TDF-ID-00008][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00009.

5.6 - `./Rules/deleted/IC-TDF_ID_00009.sch`

Rule Description

[IC-TDF-ID-00009][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00010.

5.7 - `./Rules/deleted/IC-TDF_ID_00010.sch`

Rule Description

[IC-TDF-ID-00010][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00011.

5.8 - **./Rules/deleted/IC-TDF_ID_00011.sch**

Rule Description

[IC-TDF-ID-00011][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00012.

5.9 - **./Rules/deleted/IC-TDF_ID_00012.sch**

Rule Description

[IC-TDF-ID-00012][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00013.

5.10 - **./Rules/deleted/IC-TDF_ID_00013.sch**

Rule Description

[IC-TDF-ID-00013][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00014.

5.11 - **./Rules/deleted/IC-TDF_ID_00014.sch**

Rule Description

[IC-TDF-ID-00014][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00015.

5.12 - **./Rules/deleted/IC-TDF_ID_00015.sch**

Rule Description

[IC-TDF-ID-00015][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00016.

5.13 - **./Rules/deleted/IC-TDF_ID_00020.sch**

Rule Description

[IC-TDF-ID-00020][Error] Removed in V3.

5.14 - **./Rules/deleted/IC-TDF_ID_00021.sch**

Rule Description

[IC-TDF-ID-00021][Error] Removed in V3.

5.15 - **./Rules/deleted/IC-TDF_ID_00022.sch**

Rule Description

[IC-TDF-ID-00022][Error] Removed in V3.

5.16 - **./Rules/deleted/IC-TDF_ID_00023.sch**

Rule Description

[IC-TDF-ID-00023][Error] Removed in V3.

5.17 - **./Rules/deleted/IC-TDF_ID_00024.sch**

Rule Description

[IC-TDF-ID-00024][Error] Removed in V3.

5.18 - **./Rules/deleted/IC-TDF_ID_00025.sch**

Rule Description

[IC-TDF-ID-00025][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00002.

5.19 - **./Rules/deleted/IC-TDF_ID_00032.sch**

Rule Description

[IC-TDF-ID-00032][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00003.

5.20 - **./Rules/deleted/IC-TDF_ID_00035.sch**

Rule Description

[IC-TDF-ID-00035][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00017.

5.21 - **./Rules/deleted/IC-TDF_ID_00037.sch**

Rule Description

[IC-TDF-ID-00037][Error] Removed in V2019-MAR due to ARH and NTK consolidation into ISM.

5.22 - **./Rules/deleted/IC-TDF_ID_00038.sch**

Rule Description

[IC-TDF-ID-00038][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00018.

5.23 - **./Rules/deleted/IC-TDF_ID_00039.sch**

Rule Description

[IC-TDF-ID-00039][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00004.

5.24 - **./Rules/deleted/IC-TDF_ID_00040.sch**

Rule Description

[IC-TDF-ID-00040][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00019.

5.25 - **./Rules/deleted/IC-TDF_ID_00041.sch**

Rule Description

[IC-TDF-ID-00041][Error] Removed in V2021-NOV and migrated to BASE-TDF-ID-00020.

5.26 - **./Rules/deleted/IC-TDF_ID_00047.sch**

Rule Description

[IC-TDF-ID-00047][Error] Removed in V2019-MAR due to ARH and NTK consolidation into ISM.

5.27 - **./Rules/deleted/IC-TDF_ID_00048.sch**

Rule Description

[IC-TDF-ID-00048][Error] Removed in V2019-MAR due to ARH and NTK consolidation into ISM.