



Guide to Schematron Rules and Patterns

BASE-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/BASE-TDF_ID_00001.sch	3
2.2 - ../Rules/BASE-TDF_ID_00002.sch	4
2.3 - ../Rules/BASE-TDF_ID_00003.sch	5
2.4 - ../Rules/BASE-TDF_ID_00004.sch	6
2.5 - ../Rules/BASE-TDF_ID_00005.sch	7
2.6 - ../Rules/BASE-TDF_ID_00006.sch	8
2.7 - ../Rules/BASE-TDF_ID_00007.sch	9
2.8 - ../Rules/BASE-TDF_ID_00008.sch	10
2.9 - ../Rules/BASE-TDF_ID_00009.sch	11
2.10 - ../Rules/BASE-TDF_ID_00010.sch	12
2.11 - ../Rules/BASE-TDF_ID_00011.sch	13
2.12 - ../Rules/BASE-TDF_ID_00012.sch	14
2.13 - ../Rules/BASE-TDF_ID_00013.sch	15
2.14 - ../Rules/BASE-TDF_ID_00014.sch	16
2.15 - ../Rules/BASE-TDF_ID_00015.sch	17
2.16 - ../Rules/BASE-TDF_ID_00016.sch	18
2.17 - ../Rules/BASE-TDF_ID_00017.sch	19
2.18 - ../Rules/BASE-TDF_ID_00018.sch	20
2.19 - ../Rules/BASE-TDF_ID_00019.sch	21
2.20 - ../Rules/BASE-TDF_ID_00020.sch	22
2.21 - ../Rules/BASE-TDF_ID_00021.sch	23
2.22 - ../Rules/BASE-TDF_ID_00022.sch	24
2.23 - ../Rules/BASE-TDF_ID_00023.sch	25
2.24 - ../Rules/BASE-TDF_ID_00024.sch	26
Chapter 3 - Abstract Patterns	27
3.1 - ../Lib/ValidateValidationEnvCVE.sch	28
3.2 - ../Lib/ValidateValidationEnvSchema.sch	29
Chapter 4 - Schematron Schema	30
4.1 - ../BASE-TDF_XML.sch	31
Chapter 5 - Removed Rules	33

Chapter 1 - Introduction

1.1 - Purpose

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

BASE-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 BASE-TDF's Schematron rules.

Chapter 2 - Rules

All of the numbered Rules for BASE-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 BASE-TDF-ID-XXXXX, with rule files named BASE-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/BASE-TDF_ID_00001.sch

Rule Description

[BASE-TDF-ID-00001][Warning] tdf:version attribute SHOULD be specified as version 202111 (Version:2021-NOV) 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(-.{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="BASE-TDF-ID-00001">
    <sch:rule id="BASE-TDF-ID-00001-R1" context="*[@tdf:version]">
        <sch:assert test="matches(@tdf:version, '^202111(-.{1,23})?$')"
            flag="warning"
            role="warning">[BASE-TDF-ID-00001][Warning] tdf:version attribute SHOULD be specified as version 202111 (Version:2021-NOV) with an optional extension. Found:
    <sch:value-of select="@tdf:version"/>
        </sch:assert>
    </sch:rule>
</sch:pattern>
```

2.2 - ../Rules/BASE-TDF_ID_00002.sch

Rule Description

[BASE-TDF-ID-00002][Error] Attribute @appliesToState is only allowed when TDO payload attribute @isEncrypted equals "true". Human Readable: Handling Statement state applicability can only be defined when an encrypted payload is present.

Code Description

If attribute @appliesToState is defined, this rule ensures that there is a payload element with attribute isEncrypted set to 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="BASE-TDF-ID-00002">
    <sch:rule id="BASE-TDF-ID-00002-R1"
              context="tdf:TrustedDataObject[tdf:HandlingAssertion/@tdf:appliesToState]">
        <sch:assert test=".*@tdf:isEncrypted = true()" flag="error" role="error">[BASE-TDF-ID-00002][Error] Attribute @appliesToState is only allowed when TDO payload
attribute @isEncrypted equals "true". Human Readable: Handling Statement state applicability can only be defined when an encrypted payload is present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.3 - ../Rules/BASE-TDF_ID_00003.sch

Rule Description

[BASE-TDF-ID-00003][Error] Attribute @appliesToState is only allowed when TDO statement attribute @isEncrypted equals "true". Human Readable: StatementMetadata state applicability can only be defined when an encrypted statement is present.

Code Description

If attribute @appliesToState is defined, this rule ensures that there is a statement element with attribute isEncrypted set to 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="BASE-TDF-ID-00003">
    <sch:rule id="BASE-TDF-ID-00003-R1"
        context="tdf:TrustedDataObject/tdf:Assertion[tdf:StatementMetadata/@tdf:appliesToState]">
        <sch:assert test=".*@tdf:isEncrypted = true()" flag="error" role="error">[BASE-TDF-ID-00003][Error] Attribute @appliesToState is only allowed when TDO statement
attribute @isEncrypted equals "true". Human Readable: StatementMetadata state applicability can only be defined when an encrypted statement is present.</sch:assert>
        </sch:rule>
    </sch:pattern>
```


2.4 - ../Rules/BASE-TDF_ID_00004.sch

Rule Description

[BASE-TDF-ID-00004][Error] Attribute @appliesToState is only allowed on HandlingAssertions with scope PAYL. Human Readable: Only Handling Assertions with scope PAYL can use the appliesToState attribute because the attribute indicates the state (encrypted or unencrypted) of the payload to which the assertion applies.

Code Description

If attribute @appliesToState is defined on a handlingAssertion, this rule ensures that handlingAssertion has scope PAYL

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="BASE-TDF-ID-00004">
    <sch:rule id="BASE-TDF-ID-00004-R1"
        context="tdf:TrustedDataObject/tdf:HandlingAssertion[@tdf:appliesToState]">
        <sch:assert test="@tdf:scope = 'PAYL'" flag="error" role="error">[BASE-TDF-ID-00004][Error] Attribute @appliesToState is only allowed with HandlingAssertions of scope
PAYL Human Readable: Only Handling Assertions with scope PAYL can use the appliesToState attribute because the attribute indicates the state (encrypted or unencrypted) of the payload to which
the assertion applies.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.5 - ../Rules/BASE-TDF_ID_00005.sch

Rule Description

[BASE-TDF-ID-00005][Error] All attributes in the TDF namespace MUST contain a non-whitespace value. Human Readable: All attributes in the TDF namespace must specify a value.

Code Description

For all attributes in the tdf namespace, this rule ensures that each contains a non-whitespace value.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00005">
    <sch:rule id="BASE-TDF-ID-00005-R1" context="*[@tdf:*]">
        <sch:assert test="every $attribute in @tdf:* satisfies normalize-space(string($attribute))"
                    flag="error"
                    role="error">[BASE-TDF-ID-00005][Error] All attributes in the TDF namespace must specify a value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.6 - ../Rules/BASE-TDF_ID_00006.sch

Rule Description

[BASE-TDF-ID-00006][Error] If the root element is TrustedDataObject, then it must specify attribute version. Human Readable: If TrustedDataObject is the root element, then it must declare a TDF version to which it complies.

Code Description

For a tdf:TrustedDataObject element that is a root element, this rule ensures that it specifies attribute tdf:version.

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="BASE-TDF-ID-00006">
    <sch:rule id="BASE-TDF-ID-00006-R1" context="/tdf:TrustedDataObject">
        <sch:assert test="@tdf:version" flag="error" role="error">[BASE-TDF-ID-00006][Error] If TrustedDataObject is the root element, then it must declare a TDF version to
which it complies.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.7 - ../Rules/BASE-TDF_ID_00007.sch

Rule Description

[BASE-TDF-ID-00007][Error] For any child element of TrustedDataObject, the only allowable tokens for attribute scope are [PAYL], [TDO], or [EXPLICIT]. Human Readable: Scopes defined within a TrustedDataObject must refer to the payload, the entire TrustedDataObject, the combination of the payload and the entire TrustedDataObject, or be explicitly defined.

Code Description

For the scope attribute specified on any child element of TrustedDataObject, this rule ensures that the value only contains the tokens [PAYL], [TDO], or [EXPLICIT].

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="BASE-TDF-ID-00007">
  <sch:rule id="BASE-TDF-ID-00007-R1" context="tdf:TrustedDataObject/*[@tdf:scope]">
    <sch:assert test="util:containsOnlyTheTokens(@tdf:scope, ('PAYL', 'TDO', 'EXPLICIT'))"
      flag="error"
      role="error">[BASE-TDF-ID-00007][Error] For any child element of TrustedDataObject, the only allowable tokens for attribute scope are [PAYL], [TDO], or
[EXPLICIT]. Human Readable: Scopes defined within a TrustedDataObject must refer to the payload, the entire TrustedDataObject, the combination of the payload and the entire TrustedDataObject,
or be explicitly defined.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.8 - ../Rules/BASE-TDF_ID_00008.sch

Rule Description

[BASE-TDF-ID-00008][Error] For any child assertion of TrustedDataCollection, the only allowable tokens for attribute scope are [TDC], [DESC_PAYL], [DESC_TDO], [TDC_MEMBER], or [EXPLICIT]. Human Readable: Scopes defined within a TrustedDataCollection must refer to the descendent TDOs (the list of TDOs), the descendent Payloads, a TDC Member, the entire TrustedDataCollection, or be explicitly defined.

Code Description

For the scope attribute specified on any child element of TrustedDataCollection, this rule ensures that the value only contains the tokens [TDC], [DESC_PAYL], [DESC_TDO], [TDC_MEMBER], or [EXPLICIT].

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="BASE-TDF-ID-00008">
  <sch:rule id="BASE-TDF-ID-00008-R1"
    context="tdf:TrustedDataCollection/*[@tdf:scope]">
    <sch:assert test="util:containsOnlyTheTokens(@tdf:scope, ('TDC', 'DESC_PAYL', 'DESC_TDO', 'TDC_MEMBER', 'EXPLICIT'))"
      flag="error"
      role="error">[BASE-TDF-ID-00008][Error] For any child element of TrustedDataCollection, the only allowable tokens for attribute scope are [TDC],
[DESC_PAYL], [DESC_TDO], [TDC_MEMBER], or [EXPLICIT]. Human Readable: Scopes defined within a TrustedDataCollection must refer to the descendent TDOs (the list of TDOs), the descendent
Payloads, a TDC Member, the entire TrustedDataCollection, or be explicitly defined.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.9 - ../Rules/BASE-TDF_ID_00009.sch

Rule Description

[BASE-TDF-ID-00009][Error] The use of EXPLICIT scope is not currently allowed. Key questions regarding the functionality of Binding within EXPLICIT scope are still being defined. The rest of the rules/structure relating to EXPLICIT scope are included in the spec to give the community an idea of how these rules/structures will be defined. If there is a use-case which requires EXPLICIT scope, please send an email to ic-standards-support@odni.gov so that the use-case can be incorporated while defining the behavior of EXPLICIT scope.

Code Description

For any element which specifies attribute scope containing [EXPLICIT], instantly fail because EXPLICIT scope is currently not supported.

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="BASE-TDF-ID-00009">
    <sch:rule id="BASE-TDF-ID-00009-R1"
        context="*[util:containsAnyOfTheTokens(@tdf:scope, ('EXPLICIT'))]">
        <sch:assert test="false()" flag="error" role="error">[BASE-TDF-ID-00009][Error] The use of EXPLICIT scope is not currently allowed. Key questions regarding the
functionality of Binding within EXPLICIT scope are still being defined. The rest of the rules/structure relating to EXPLICIT scope are included in the spec to give the community an idea of how
these rules/structures will be defined. If there is a use-case which requires EXPLICIT scope, please send an email to ic-standards-support@odni.gov so that the use-case can be incorporated
while defining the behavior of EXPLICIT scope.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.10 - ../Rules/BASE-TDF_ID_00010.sch

Rule Description

[BASE-TDF-ID-00010][Error] For element Binding, if element BoundValueList is specified, then element SignatureValue must not specify attribute includesStatementMetadata. Human Readable: If BoundValueList is present, then it will explicitly specify includesStatementMetadata for each BoundValue and therefore attribute includesStatementMetadata on the SignatureValue is not applicable.

Code Description

For element Binding which specifies BoundValueList, this rule ensures that element SignatureValue does not specify attribute includesStatementMetadata.

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="BASE-TDF-ID-00010">
  <sch:rule id="BASE-TDF-ID-00010-R1" context="tdf:Binding[tdf:BoundValueList]">
    <sch:assert test="not(tdf:SignatureValue/@tdf:includesStatementMetadata)"
              flag="error"
              role="error">[BASE-TDF-ID-00010][Error] For element Binding, if element BoundValueList is specified, then element SignatureValue must not specify attribute
includesStatementMetadata. Human Readable: If BoundValueList is present, then it will explicitly specify includesStatementMetadata for each BoundValue and therefore attribute
includesStatementMetadata on the SignatureValue is not applicable.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.11 - .//Rules/BASE-TDF_ID_00011.sch

Rule Description

[BASE-TDF-ID-00011][Error] For element Binding, if element BoundValueList is not specified, then element SignatureValue must specify attribute includesStatementMetadata. Human Readable: If BoundValueList is not present, then SignatureValue must indicate whether or not to include the StatementMetadata of all Assertions included in the binding.

Code Description

For element Binding that does not have child element BoundValueList, this rule ensures that child element SignatureValue specifies attribute includesStatementMetadata.

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="BASE-TDF-ID-00011">
    <sch:rule id="BASE-TDF-ID-00011-R1"
              context="tdf:Binding[not(tdf:BoundValueList)]">
        <sch:assert test="tdf:SignatureValue/@tdf:includesStatementMetadata"
                    flag="error"
                    role="error">[BASE-TDF-ID-00011][Error] For element Binding, if element BoundValueList is not specified, then element SignatureValue must specify attribute
includesStatementMetadata. Human Readable: If BoundValueList is not present, then SignatureValue must indicate whether or not to include the StatementMetadata of all Assertions included in the
binding.</sch:assert>
    </sch:rule>
</sch:pattern>
```


2.12 - .//Rules/BASE-TDF_ID_00012.sch

Rule Description

[BASE-TDF-ID-00012][Error] For all BoundValue or Reference elements within a TrustedDataObject, idRef attribute values must reference the id value of a descendant of the same TrustedDataObject that contains the Reference or BoundValue element. Human Readable: Assertions and HandlingAssertions within a TrustedDataObject must reference elements local to that TrustedDataObject.

Code Description

For element TrustedDataObject, this rule ensures each attribute @idRef value has matching @id value in the same 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="BASE-TDF-ID-00012">
  <sch:rule id="BASE-TDF-ID-00012-R1" context="tdf:TrustedDataObject">
    <sch:let name="ids" value="//@tdf:id"/>
    <sch:let name="externalIdRefs"
      value="for $idRef in .//@tdf:idRef return if($idRef = $ids) then null else $idRef"/>
    <sch:assert test="count($externalIdRefs) = 0" flag="error" role="error">[BASE-TDF-ID-00012][Error] For all BoundValue or Reference elements within a TrustedDataObject,
idRef attribute values must reference the id value of a descendant of the same TrustedDataObject that contains the Reference or BoundValue element. Human Readable: Assertions and
HandlingAssertions within a TrustedDataObject must reference elements local to that TrustedDataObject. The following idRefs reference elements outside of this TrustedDataObject: (
    <sch:value-of select="for $externalRef in $externalIdRefs return concat(string($externalRef), ', ')" />).
  </sch:assert>
  </sch:rule>
</sch:pattern>
```

2.13 - .//Rules/BASE-TDF_ID_00013.sch

Rule Description

[BASE-TDF-ID-00013][Error] For any element which specifies attribute scope containing [EXPLICIT], then element Binding/BoundValueList or element ReferenceList must be specified. Human Readable: Assertions with explicit scope require either a BoundValueList or a ReferenceList to identify the elements for which the assertion applies.

Code Description

For elements which specify attribute scope with a value of [EXPLICIT], this rule ensures that element Binding/BoundValueList or ReferenceList is specified.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00013">
    <sch:rule id="BASE-TDF-ID-00013-R1"
        context="*[normalize-space(string(@tdf:scope)) = 'EXPLICIT']">
        <sch:assert test="tdf:Binding/tdf:BoundValueList or tdf:ReferenceList"
            flag="error"
            role="error">[BASE-TDF-ID-00013][Error] For any element which specifies attribute scope containing [EXPLICIT], then element Binding/BoundValueList or
element ReferenceList must be specified. Human Readable: Assertions with explicit scope require either a BoundValueList or a ReferenceList to identify the elements for which the assertion
applies.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.14 - ./Rules/BASE-TDF_ID_00014.sch

Rule Description

[BASE-TDF-ID-00014][Error] Elements ReferenceList and BoundValueList are currently not allowed. Key questions regarding the functionality of granular references and granular binding are still being defined. The rest of the rules/structure relating to these elements are included in the spec to give the community an idea of how these rules/structures will be defined. If there is a use-case which requires granular references or granular binding, please send an email to ic-standards-support@odni.gov so that the use-case can be incorporated while defining the behavior and rules.

Code Description

Elements ReferenceList and BoundValueList are not allowed in this version. This rule will in the future require that elements which specify element ReferenceList or Binding/BoundValueList have attribute scope is specified with a value of [EXPLICIT].

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="BASE-TDF-ID-00014">
  <sch:rule id="BASE-TDF-ID-00014-R1"
    context="tdf:ReferenceList | tdf:Binding/tdf:BoundValueList">
    <sch:assert test="false()" flag="error" role="error">[BASE-TDF-ID-00014][Error] Elements ReferenceList and BoundValueList are currently not allowed. Key questions
regarding the functionality of granular references and granular binding are still being defined. The rest of the rules/structure relating to these elements are included in the spec to give the
community an idea of how these rules/structures will be defined. If there is a use-case which requires granular references or granular binding, please send an email to ic-standards-
support@odni.gov so that the use-case can be incorporated while defining the behavior and rules.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.15 - ../Rules/BASE-TDF_ID_00015.sch

Rule Description

[BASE-TDF-ID-00015][Error] If EncryptionInformation is specified, then the data it refers to must be labeled as encrypted. (Assertion Statement or TrustedDataObject Payload).

Code Description

This rule ensures that the following sibling of EncryptionInformation, the Payload or Assertion Statement, has the encrypted attribute set to 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="BASE-TDF-ID-00015">
    <sch:rule id="BASE-TDF-ID-00015-R1"
        context="tdf:EncryptionInformation[parent::tdf:Assertion] | tdf:EncryptionInformation[parent::tdf:TrustedDataObject]">
        <sch:assert test="following-sibling::tdf:*[@tdf:isEncrypted=true()]"
            flag="error"
            role="error">[BASE-TDF-ID-00015][Error] If EncryptionInformation is specified, then the data it refers to must be labeled as encrypted. (Assertion Statement
or TrustedDataObject Payload).</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.16 - ../Rules/BASE-TDF_ID_00016.sch

Rule Description

[BASE-TDF-ID-00016][Error] If data is labeled as encrypted, then EncryptionInformation must be specified. (Assertion Statement or TrustedDataObject Payload).

Code Description

This rule ensures that the previous sibling of the Statement or Payload marked with the encrypted attribute set to true is EncryptionInformation.

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="BASE-TDF-ID-00016">
    <sch:rule id="BASE-TDF-ID-00016-R1" context="tdf:*[@tdf:isEncrypted=true()]">
        <sch:assert test="preceding-sibling::tdf:EncryptionInformation"
                    flag="error"
                    role="error">[BASE-TDF-ID-00016][Error] If data is labeled as encrypted, then EncryptionInformation must be specified. (Assertion Statement or
TrustedDataObject Payload).</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.17 - .//Rules/BASE-TDF_ID_00017.sch

Rule Description

[BASE-TDF-ID-00017][Error] For any handling assertion child element of TrustedDataCollection, the only allowable token for attribute scope is [TDC]. Human Readable: Scopes defined within a TrustedDataCollection Handling Assertion must refer to entire TrustedDataCollection.

Code Description

For the scope attribute specified on handlingAssertion child elements of TrustedDataCollection, make sure that the value only contains the tokens [TDC].

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="BASE-TDF-ID-00017">
    <sch:rule id="BASE-TDF-ID-00017-R1"
        context="tdf:TrustedDataCollection/tdf:HandlingAssertion">
        <sch:assert test="util:containsOnlyTheTokens(@tdf:scope, ('TDC'))"
            flag="error"
            role="error">[BASE-TDF-ID-00017][Error] For any child handlingAssertion of TrustedDataCollection, the only allowable tokens for attribute scope is [TDC].
Human Readable: Scopes defined within a TrustedDataCollection Handling Assertion must refer to entire TrustedDataCollection.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.18 - ../Rules/BASE-TDF_ID_00018.sch

Rule Description

[BASE-TDF-ID-00018][Error] For the Binding element, every Signer element must specify the issuer attribute and either the serial or subject attribute.

Code Description

This rule checks that for each occurrence of tdf:Signer that @tdf:issuer and either @tdf:subject or @tdf:serial is specified.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00018">
    <sch:rule id="BASE-TDF-ID-00018-R1" context="tdf:Binding/tdf:Signer">
        <sch:assert test="@tdf:issuer and (@tdf:serial or @tdf:subject)"
                    flag="error"
                    role="error">[BASE-TDF-ID-00018][Error] For the Binding element, every Signer element must specify the issuer attribute and either the serial or subject
attribute.</sch:assert>
    </sch:rule>
</sch:pattern>
```

2.19 - ../Rules/BASE-TDF_ID_00019.sch

Rule Description

[BASE-TDF-ID-00019][Error] If there are more than one EncryptionInformation elements specified in any one EncryptionInformation Group than @tdf:sequenceNum must also be specified.

Code Description

This rule checks that if there are more than one tdf:EncryptionInformation in any encryption group (if it has siblings) then it checks that a tdf:sequenceNum attribute is present on the EncryptionInformation 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="BASE-TDF-ID-00019">
    <sch:rule id="BASE-TDF-ID-00019-R1"
        context="tdf:EncryptionInformation[count((preceding-sibling::tdf:EncryptionInformation, following-sibling::tdf:EncryptionInformation))>0]">
        <sch:assert test="@tdf:sequenceNum" flag="error" role="error">[BASE-TDF-ID-00019][Error] If there are more than one EncryptionInformation elements specified in any one
EncryptionInformation Group than @tdf:sequenceNum must also be specified.</sch:assert>
    </sch:rule>
</sch:pattern>
```


2.20 - .//Rules/BASE-TDF_ID_00020.sch

Rule Description

[BASE-TDF-ID-00020][Error] All sequenceNum attributes in an EncryptionInformation Group must be sequential, incrementing by 1, starting with the number 1, and contain no duplicates.

Code Description

This rule triggers on the first EncryptionInformation element for each EncryptionInformation Group that has more than 1 EncryptionInformation element then checks that the sequenceNum attributes are numerically sequential by 1 starting from 1. A list, named \$nums, is created containing the value of each sequenceNum attribute within the group. If the total number of items in \$nums does not equal the number of distinct values in \$nums, then a duplicate exists return false. Otherwise, ensure that each number from 1 to N, where N is the number of items in \$nums, is contained within \$nums. If each number is contained, then return true. Otherwise, false.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00020"><!--This rule triggers on the first EncryptionInformation element for each EncryptionInformation Group
that has more than 1 EncryptionInformation element then checks that the sequenceNum attributes
are numerically sequential by 1 starting from 1. The test uses the mathematical formula
(1+last*count)/2 = sum(1...count) derived from the formula
(first+last)*count/2 = sum(first...last) where first is replaced by 1 and last
is replaced by count. This works by assuming first=1 then it must
be true that last=count.-->

<sch:rule id="BASE-TDF-ID-00020-R1"
      context="tdf:EncryptionInformation[count(following-sibling::tdf:EncryptionInformation)>0][1]">
  <sch:let name="nums"
    value="for $encInfo in (., following-sibling::tdf:EncryptionInformation) return number($encInfo/@tdf:sequenceNum)"/>
  <sch:assert test="(count(distinct-values($nums)) = count($nums) and (every $index in 1 to count($nums) satisfies index-of($nums, $index)))"
    flag="error"
    role="error">[BASE-TDF-ID-00020][Error] All sequenceNum attributes in an EncryptionInformation Group must be sequential, incrementing by 1, starting with
the number 1, and contain no duplicates.</sch:assert>
  </sch:rule>
</sch:pattern>
```

2.21 - ../Rules/BASE-TDF_ID_00021.sch

Rule Description

[BASE-TDF-ID-00021][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="BASE-TDF-ID-00021" 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="'BASE-TDF-ID-00021'"/>
</sch:pattern>
```

2.22 - ../Rules/BASE-TDF_ID_00022.sch

Rule Description

[BASE-TDF-ID-00022][Error] For any tdf:TrustedDataCollection or tdf:TrustedDataObject containing references to any sfhashv (urn:us:gov:ic:sf:hashverification) elements, that tdf:TrustedDataCollection or tdf:TrustedDataObject must have the @sf:DESVersion attribute.

Code Description

For any tdf:TrustedDataCollection or tdf:TrustedDataObject containing references to any sfhashv (urn:us:gov:ic:sf:hashverification) elements, that tdf:TrustedDataCollection or tdf:TrustedDataObject must have the @sf:DESVersion attribute.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00022">
  <sch:rule id="BASE-TDF-ID-00022-R1"
    context="tdf:TrustedDataCollection[../sfhashv:* | ../*/@sfhashv:*] | tdf:TrustedDataObject[../sfhashv:* | ../*/@sfhashv:*]">
    <sch:assert test="../@sf:DESVersion" flag="error" role="error">[BASE-TDF-ID-00022][Error] For any tdf:TrustedDataCollection or tdf:TrustedDataObject containing
references to any sfhashv (urn:us:gov:ic:sf:hashverification) elements, that tdf:TrustedDataCollection or tdf:TrustedDataObject must have the @sf:DESVersion attribute.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

2.23 - ../Rules/BASE-TDF_ID_00023.sch

Rule Description

[BASE-TDF-ID-00023][Warning] If a TDO contains a cryptographic binding for a reference value payload, it should also contain the hash to verify that binding. If the reference value payload is encrypted, then the hash can be the encoded and/or decoded hash. If the reference value payload is not encrypted, then the hash is expected to be the decoded hash.

Code Description

A tdf:TrustedDataObject that contains tdf:Binding and tdf:ReferenceValuePayload should contain sfhashv:ContentEncodedHashVerification or sfhashv:ContentDecodedHashVerification if @tdf:isEncrypted = 'true' or just sfhashv:ContentDecodedHashVerification if @tdf:isEncrypted did not exist or @tdf:isEncrypted = 'false'.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00023">
    <sch:rule id="BASE-TDF-ID-00023-R1"
        context="tdf:TrustedDataObject[../tdf:Binding and ../tdf:ReferenceValuePayload]">
        <sch:assert test="if (../*/@tdf:isEncrypted and ../*/@tdf:isEncrypted = 'true') then ../sfhashv:ContentEncodedHashVerification or ../sfhashv:ContentDecodedHashVerification else ../sfhashv:ContentDecodedHashVerification"
            flag="warning"
            role="warning">[BASE-TDF-ID-00023][Warning] If a TDO contains a cryptographic binding for a reference value payload, it should also contain the hash to
verify that binding. If the reference value payload is encrypted, then the hash can be the encoded and/or decoded hash. If the reference value payload is not encrypted, then the hash is
expected to be the decoded hash.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

2.24 - ../Rules/BASE-TDF_ID_00024.sch

Rule Description

[BASE-TDF-ID-00024][Warning] If a TDF assertion contains a cryptographic binding for a reference statement, it should also contain the hash to verify that binding. If the reference statement is encrypted, then the hash can be the encoded and/or decoded hash. If the reference statement is not encrypted, then the hash is expected to be the decoded hash.

Code Description

A tdf:Assertion that contains tdf:Binding and tdf:ReferenceStatement should contain sfhashv:ContentEncodedHashVerification or sfhashv:ContentDecodedHashVerification if @tdf:isEncrypted = 'true' or just sfhashv:ContentDecodedHashVerification if @tdf:isEncrypted did not exist or @tdf:isEncrypted = 'false'.

Schematron Code

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

<sch:pattern id="BASE-TDF-ID-00024">
    <sch:rule id="BASE-TDF-ID-00024-R1"
        context="tdf:Assertion[../tdf:Binding and ../tdf:ReferenceStatement]">
        <sch:assert test="if (../*/@tdf:isEncrypted and ../*/@tdf:isEncrypted = 'true') then ../sfhashv:ContentEncodedHashVerification or ../sfhashv:ContentDecodedHashVerification else ../sfhashv:ContentDecodedHashVerification"
            flag="warning"
            role="warning">[BASE-TDF-ID-00024][Warning] If a TDF assertion contains a cryptographic binding for a reference statement, it should also contain the hash
to verify that binding. If the reference statement is encrypted, then the hash can be the encoded and/or decoded hash. If the reference statement is not encrypted, then the hash is expected to
be the decoded hash.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

Chapter 3 - Abstract Patterns

All of the Abstract Patterns for BASE-TDF 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

```
<!--
  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>
```

3.2 - ./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 >= $MinVersion"
               flag="error"
               role="error">[
  <sch:value-of select="$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 BASE-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 - ./BASE-TDF_XML.sch

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 xmlns:sfhashv="urn:us:gov:ic:sf:hashverification" queryBinding="xslt2">
    <sch:ns uri="urn:us:gov:ic:tdf" prefix="tdf"/>
    <sch:ns uri="urn:us:gov:ic:ism" prefix="ism"/>
    <sch:ns uri="urn:us:gov:ic:sf:hashverification" prefix="sfhashv"/>
    <sch:ns uri="urn:us:gov:ic:sf" prefix="sf"/>
    <sch:ns prefix="util" uri="urn:us:gov:ic:tdf:xsl:util"/>
    <!--*****-->
<!-- (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>
<!-- ***** -->
<!-- * Abstract Rule and Pattern Includes * -->
<!-- ***** -->

<sch:include href="./Lib/ValidateValidationEnvSchema.sch"/>
<sch:include href="./Lib/ValidateValidationEnvCVE.sch"/>
```

```
<!--*****-->
<!-- (U) BASE-TDF Phases -->
<!--*****-->
<!--*****-->
<!-- (U) BASE-TDF ID Rules -->
<!--*****-->
<!--(U) -->

<sch:include href="./Rules/BASE-TDF_ID_00001.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00002.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00003.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00004.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00005.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00006.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00007.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00008.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00009.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00010.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00011.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00012.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00013.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00014.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00015.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00016.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00017.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00018.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00019.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00020.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00021.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00022.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00023.sch"/>
    <sch:include href="./Rules/BASE-TDF_ID_00024.sch"/>
    <!--*****-->
<!-- (U) BASE-TDF Phases -->
<!--*****-->
</sch:schema>

<!--UNCLASSIFIED-->
```

Chapter 5 - Removed Rules

There are no rules that have been removed for BASE-TDF.