Guide to Schematron Rules and Patterns

IC-TDF Schematron Guide

Version 1

Date of Draft Release: 21 August 2012

17 July 2012

Table of Contents

Chapter 1 - Introduction

1.1 - Purpose

(U) The following documentation is informative. The actual Schematron files are the normative record. This documentation is generated from the Schematron files via XSLT it may be missing some file or pieces of a file but whatever is here other than titles came from the original file.

(U) It is envisioned that this will be a useful resource to search and read but for questions and debates the source files should be consulted.

(U) 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 strongly influence a rule but is not actually a numbered rule.

Chapter 2 - Rules

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

2.1 - .//Rules/IC-TDF_ID_00001.sch

Rule Description: IC-TDF-ID-00001 [IC-TDF-ID-00001][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, we make sure 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="IC-TDF-ID-00001">


                  <sch:rule context="*[@tdf:*]">
                     <sch:assert test="every $attribute in @tdf:*
satisfies               normalize-space(string($attribute))"
                                 flag="error">
            [IC-TDF-ID-00001][Error] All attributes in the TDF namespace must
specify a value.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.2 - .//Rules/IC-TDF_ID_00002.sch

Rule Description: IC-TDF-ID-00002 [IC-TDF-ID-00002][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, we make sure 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="IC-TDF-ID-00002">


                  <sch:rule context="/tdf:TrustedDataObject">
                     <sch:assert test="@tdf:version" flag="error">
            [IC-TDF-ID-00002][Error] If TrustedDataObject is the root
element, then it must declare a TDF version to which it complies.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.3 - .//Rules/IC-TDF_ID_00003.sch

Rule Description: IC-TDF-ID-00003 [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, we make sure 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 context="tdf:TrustedDataObject">
                          <sch:assert
test="count(child::tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scop
e, ('PAYL'))])&gt;= 1"
                                    flag="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>
```

2.4 - .//Rules/IC-TDF_ID_00004.sch

Rule Description: IC-TDF-ID-00004 [IC-TDF-ID-00004][Error] For element TrustedDataObject, there must be exactly one element HandlingAssertion which specifies attribute scope containing [TDO]. Human Readable: There must exist a single handling marking for the entire TrustedDataObject.

Code Description: For element TrustedDataObject, we make sure that the count of HandlingAssertion elements which specify attribute scope containing [TDO] 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 context="tdf:TrustedDataObject">
                          <sch:assert
test="count(child::tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scop
e, ('TDO'))])= 1"
                                    flag="error">
            [IC-TDF-ID-00004][Error] For element TrustedDataObject, there
must be
            exactly one element HandlingAssertion which specifies attribute
scope
            containing [TDO].

            Human Readable: There must exist a single handling marking for
the entire TrustedDataObject.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.5 - .//Rules/IC-TDF_ID_00005.sch

Rule Description: IC-TDF-ID-00005 [IC-TDF-ID-00005][Error] For element TrustedDataCollection, there must be exactly one element HandlingAssertion which specifies attribute scope containing [TDC]. Human Readable: There must exist a single handling marking for the entire TrustedDataCollection.

Code Description: For element TrustedDataCollection, we make sure that the count of HandlingAssertion elements which specify attribute scope containing [TDC] 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 context="tdf:TrustedDataCollection">
                          <sch:assert
test="count(child::tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scop
e, ('TDC'))])= 1"
                                    flag="error">
            [IC-TDF-ID-00005][Error] For element TrustedDataCollection, there
must be
            exactly one element HandlingAssertion which specifies attribute
scope
            containing [TDC].

            Human Readable: There must exist a single handling marking for the
            entire TrustedDataCollection.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.6 - .//Rules/IC-TDF_ID_00006.sch

Rule Description: IC-TDF-ID-00006 [IC-TDF-ID-00006][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, we make sure 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="IC-TDF-ID-00006">


                    <sch:rule context="tdf:TrustedDataObject/*[@tdf:scope]">
                         <sch:assert
test="util:containsOnlyTheTokens(@tdf:scope, ('PAYL', 'TDO', 'EXPLICIT'))"
                                  flag="error">
            [IC-TDF-ID-00006][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>
```

2.7 - .//Rules/IC-TDF_ID_00007.sch

Rule Description: IC-TDF-ID-00007 [IC-TDF-ID-00007][Error] For any child element of TrustedDataCollection, the only allowable tokens for attribute scope are [PAYL], [TDC], or [EXPLICIT]. Human Readable: Scopes defined within a TrustedDataCollection must refer to the payload (the list of TDOs), the entire TrustedDataCollection, or be explicitly defined.

Code Description: For the scope attribute specified on any child element of TrustedDataCollection, we make sure that the value only contains the tokens [PAYL], [TDC], 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="IC-TDF-ID-00007">


                     <sch:rule context="tdf:TrustedDataCollection/
*[@tdf:scope]">
                              <sch:assert
test="util:containsOnlyTheTokens(@tdf:scope, ('PAYL', 'TDC', 'EXPLICIT'))"
                                       flag="error">
            [IC-TDF-ID-00007][Error] For any child element of
TrustedDataCollection,
            the only allowable tokens for attribute scope are [PAYL], [TDC],
or [EXPLICIT].

            Human Readable: Scopes defined within a TrustedDataCollection
must refer
            to the payload (the list of TDOs), the TrustedDataCollection
itself, or
            be explicitly defined.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.8 - .//Rules/IC-TDF_ID_00008.sch

Rule Description: IC-TDF-ID-00008 [IC-TDF-ID-00008][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 you have a use-case which requires EXPLICIT scope, please send an email to datastandardssupport@ugov.gov so that we can incorporate the use-case while defining the behavior of EXPLICIT scope.

Code Description: For any element which specifies attribute scope containing [EXPLICIT], we 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="IC-TDF-ID-00008">


                     <sch:rule
context="*[util:containsAnyOfTheTokens(@tdf:scope, ('EXPLICIT'))]">
                           <sch:assert test="false()" flag="error">
            [IC-TDF-ID-00008][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 you have a use-case which requires EXPLICIT scope, please send
an
            email to datastandardssupport@ugov.gov so that we can incorporate
the
            use-case while defining the behavior of EXPLICIT scope.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.9 - .//Rules/IC-TDF_ID_00009.sch

Rule Description: IC-TDF-ID-00009 [IC-TDF-ID-00009][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, we make sure 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="IC-TDF-ID-00009">


                    <sch:rule context="tdf:Binding[tdf:BoundValueList]">
                          <sch:assert test="not(tdf:SignatureValue/
@tdf:includesStatementMetadata)" flag="error">
            [IC-TDF-ID-00009][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>
```

2.10 - .//Rules/IC-TDF_ID_00010.sch

Rule Description: IC-TDF-ID-00010 [IC-TDF-ID-00010][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, we make sure 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="IC-TDF-ID-00010">


                 <sch:rule context="tdf:Binding[not(tdf:BoundValueList)]">
                    <sch:assert test="tdf:SignatureValue/
@tdf:includesStatementMetadata" flag="error">
            [IC-TDF-ID-00010][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>
```

2.11 - .//Rules/IC-TDF_ID_00011.sch

Rule Description: IC-TDF-ID-00011 [IC-TDF-ID-00011][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, we ensure 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="IC-TDF-ID-00011">


                  <sch:rule 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">
            [IC-TDF-ID-00011][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>
```

2.12 - .//Rules/IC-TDF_ID_00012.sch

Rule Description: IC-TDF-ID-00012 [IC-TDF-ID-00012][Error] For any element which specifies attribute scope containing [EXPLICIT], then element Binding/BoundValueList or element ReferenceList must be specified. Human Readable: For explicit scope, you must use a BoundValueList or a ReferenceList to explicitly reference elements are in scope.

Code Description: For elements which specify attribute scope with a value of [EXPLICIT], we make sure 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="IC-TDF-ID-00012">


                     <sch:rule context="*[normalize-space(string(@tdf:scope))
= 'EXPLICIT']">
                           <sch:assert test="tdf:Binding/tdf:BoundValueList
or tdf:ReferenceList" flag="error">
            [IC-TDF-ID-00012][Error] For any element which specifies
attribute scope
            containing [EXPLICIT], then element Binding/BoundValueList or
            element ReferenceList must be specified.

            Human Readable: For explicit scope, you must use a BoundValueList
or
            a ReferenceList to explicitly reference elements are in scope.
        </sch:assert>
                 </sch:rule>
              </sch:pattern>
```

2.13 - .//Rules/IC-TDF_ID_00013.sch

Rule Description: IC-TDF-ID-00013 [IC-TDF-ID-00013][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 you have a use-case which requires granular references or granular binding, please send an email to datastandardssupport@ugov.gov so that we can incorporate the use-case while defining the behavior and rules.

Code Description: Elements ReferenceList and BoundValueList are not allowed in v1. 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="IC-TDF-ID-00013">


                     <sch:rule context="tdf:ReferenceList | tdf:Binding/
tdf:BoundValueList">
                              <sch:assert test="false()" flag="error">
            [IC-TDF-ID-00013][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 you have a use-case which requires granular references or
granular
            binding, please send an email to datastandardssupport@ugov.gov so
that
            we can incorporate the use-case while defining the behavior and
rules.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.14 - .//Rules/IC-TDF_ID_00014.sch

Rule Description: IC-TDF-ID-00014 [IC-TDF-ID-00014][Error] If EncryptionInformation is specified, then the data it refers to must be label as encrypted. (Assertion Statement or TrustedDataObject Payload).

Code Description: Make sure 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="IC-TDF-ID-00014">


                     <sch:rule context="tdf:EncryptionInformation">
                           <sch:assert test="following-
sibling::tdf:*[@tdf:isEncrypted=true()]" flag="error">
            [IC-TDF-ID-00014][Error] If EncryptionInformation is specified,
then
            the data it refers to must be label as encrypted. (Assertion
Statement
            or TrustedDataObject Payload).
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.15 - .//Rules/IC-TDF_ID_00015.sch

Rule Description: IC-TDF-ID-00015 [IC-TDF-ID-00015][Error] If data is label as encrypted, then EncryptionInformation must be specified. (Assertion Statement or TrustedDataObject Payload).

Code Description: Make sure 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="IC-TDF-ID-00015">


                     <sch:rule context="tdf:*[@tdf:isEncrypted=true()]">
                           <sch:assert test="preceding-
sibling::tdf:EncryptionInformation" flag="error">
            [IC-TDF-ID-00015][Error] If data is label as encrypted, then
            EncryptionInformation must be specified. (Assertion Statement
            or TrustedDataObject Payload).
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.16 - .//Rules/IC-TDF_ID_00016.sch

Rule Description: IC-TDF-ID-00016 [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: When a HandlingAssertion has scope pertaining to the entire TrustedDataObject (TDO) it must declare itself a resource level object.

Code Description: Where a HandlingAssertion exists with scope containing [TDO], ensure that its decendant 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-00016">


                     <sch:rule
context="tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope,
('TDO'))]">
                           <sch:assert
test="descendant::arh:*[@ism:resourceElement=true()]" flag="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: When a HandlingAssertion has scope pertaining to
            the entire TrustedDataObject (TDO) it must declare itself a
resource level
            object.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.17 - .//Rules/IC-TDF_ID_00017.sch

Rule Description: IC-TDF-ID-00017 [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.

Code Description: Where a HandlingAssertion exists with scope containing [TDC], ensure that its decendant 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
context="tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope,
('TDC'))]">
                           <sch:assert
test="descendant::arh:*[@ism:resourceElement=true()]" flag="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>
```

2.18 - .//Rules/IC-TDF_ID_00018.sch

Rule Description: IC-TDF-ID-00018 [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], ensure 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
context="tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope,
('TDO'))]">
                           <sch:assert
test="not(descendant::edh:ExternalEdh)" flag="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>
```

2.19 - .//Rules/IC-TDF_ID_00019.sch

Rule Description: IC-TDF-ID-00019 [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], ensure 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
context="tdf:HandlingAssertion[util:containsAnyOfTheTokens(@tdf:scope,
('TDC'))]">
                           <sch:assert
test="not(descendant::edh:ExternalEdh)" flag="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>
```

2.20 - .//Rules/IC-TDF_ID_00020.sch

Rule Description: IC-TDF-ID-00020 [IC-TDF-ID-00020][Error] For element TrustedDataObject, if element ReferenceValuePayload is specified, then there must exist two independent HandlingAssertion elements, one with attribute scope specified with a value of [TDO] and the other with attribute scope specified with a value of [PAYL]. Human Readable: If a ReferenceValuePayload is used, then the instance is not the minimal case and must specify markings for the payload and the TDO instance separately.

Code Description: For every TrustedDataObject with a ReferenceValuePayload child element, we ensure that there exists two independent HandlingAssertion elements, one of scope [TDO] and one of [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="IC-TDF-ID-00020">


                     <sch:rule
context="tdf:TrustedDataObject[tdf:ReferenceValuePayload]">
                           <sch:assert
test="not(tdf:HandlingAssertion[@tdf:scope='TDO PAYL'])" flag="error">
            [IC-TDF-ID-00020][Error] For element TrustedDataObject, if element
            ReferenceValuePayload is specified, then there must exist two
independent
            HandlingAssertion elements, one with attribute scope specified
with a value
            of [TDO] and the other with attribute scope specified with a
value of [PAYL].

            Human Readable: If a ReferenceValuePayload is used, then the
instance
            is not the minimal case and must specify markings for the payload
            and the TDO instance separately.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.21 - .//Rules/IC-TDF_ID_00021.sch

Rule Description: IC-TDF-ID-00021 [IC-TDF-ID-00021][Error] For element TrustedDataObject, if any Assertion elements are specified, then there must exist two independent HandlingAssertion elements, one with attribute scope specified with a value of [TDO] and the other with attribute scope specified with a value of [PAYL]. Human Readable: If any Assertion elements are specified, then the instance is not the minimal case and must specify markings for the payload and the TDO instance separately.

Code Description: For every TrustedDataObject with an Assertion child element, we ensure that there exists two independent HandlingAssertion elements, one of scope [TDO] and one of [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="IC-TDF-ID-00021">


                     <sch:rule context="tdf:TrustedDataObject[tdf:Assertion]">
                           <sch:assert
test="not(tdf:HandlingAssertion[@tdf:scope='TDO PAYL'])" flag="error">
            [IC-TDF-ID-00021][Error] For element TrustedDataObject, if any
Assertion
            elements are specified, then there must exist two independent
            HandlingAssertion elements, one with attribute scope specified
with a
            value of [TDO] and the other with attribute scope specified with
a
            value of [PAYL].


            Human Readable: If any Assertion elements are specified, then the
instance
            is not the minimal case and must specify markings for the payload
            and the TDO instance separately.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.22 - .//Rules/IC-TDF_ID_00022.sch

Rule Description: IC-TDF-ID-00022 [IC-TDF-ID-00022][Error] For element TrustedDataObject, if there is a single HandlingAssertion element and it contains any ism:Notice elements which specify any ISM attributes other than ism:classification with a value of [U] and ism:ownerProducer with a value of [USA], then there must exist two independent HandlingAssertion elements, one with attribute scope specified with a value of [TDO] and the other with attribute scope specified with a value of [PAYL]. Human Readable: If the single IC-EDH contains classified information within a Notice, then the instance is not the minimal case and must specify markings for the payload and the TDO instance separately.

Code Description: For TrustedDataObject elements with a single HandlingAssertion descendant element and any ism:Notice elements which specify any ISM attributes other than ism:classification with a value of [U] and ism:ownerProducer with a value of [USA], we ensure that there exists two independent HandlingAssertion elements, one of scope [TDO] and one of [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="IC-TDF-ID-00022">


                     <sch:rule context="tdf:TrustedDataObject   [count(.//
tdf:HandlingAssertion)=1]   [some $notice in tdf:HandlingAssertion//
ism:Notice satisfies    ($notice/@ism:classification and    (not($notice/
@ism:classification='U') or not($notice/@ism:ownerProducer='USA')))]">
                           <sch:assert
test="not(tdf:HandlingAssertion[@tdf:scope='TDO PAYL'])" flag="error">
            [IC-TDF-ID-00022][Error] For element TrustedDataObject, if there
is
            a single HandlingAssertion element and it contains any ism:Notice
            elements which specify any ISM attributes other than
ism:classification
            with a value of [U] and ism:ownerProducer with a value of [USA],
then
            there must exist two independent HandlingAssertion elements, one
with
            attribute scope specified with a value of [TDO] and the other
with
            attribute scope specified with a value of [PAYL].

            Human Readable: If the single IC-EDH contains classified
information
            within a Notice, then the instance is not the minimal case and
must
            specify markings for the payload and the TDO instance separately.
        </sch:assert>
                  </sch:rule>
```

```
        </sch:pattern>
```

2.23 - .//Rules/IC-TDF_ID_00023.sch

Rule Description: IC-TDF-ID-00023 [IC-TDF-ID-00023][Error] For element TrustedDataObject, if there is a single HandlingAssertion element and it contains any edh:ResponsibleEntity elements which specify any ISM attributes other than ism:classification with a value of [U] and ism:ownerProducer with a value of [USA], then there must exist two independent HandlingAssertion elements, one with attribute scope specified with a value of [TDO] and the other with attribute scope specified with a value of [PAYL]. Human Readable: If the single IC-EDH contains classified information within a ResponsibleEntity, then the instance is not the minimal case and must specify markings for the payload and the TDO instance separately.

Code Description: For TrustedDataObject elements with a single HandlingAssertion descendant element and any ism:Notice elements which specify any ISM attributes other than ism:classification with a value of [U] and ism:ownerProducer with a value of [USA], we ensure that there exists two independent HandlingAssertion elements, one of scope [TDO] and one of [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="IC-TDF-ID-00023">


                     <sch:rule context="tdf:TrustedDataObject   [count(.//
tdf:HandlingAssertion)=1]   [some $responsibleEntity in
tdf:HandlingAssertion//edh:ResponsibleEntity satisfies    ($responsibleEntity/
@ism:classification    and (not($responsibleEntity/@ism:classification='U')
or not($responsibleEntity/@ism:ownerProducer='USA')))]">
                              <sch:assert
test="not(tdf:HandlingAssertion[@tdf:scope='TDO PAYL'])" flag="error">
            [IC-TDF-ID-00023][Error] For element TrustedDataObject, if there
is
            a single HandlingAssertion element and it contains any
edh:ResponsibleEntity
            elements which specify any ISM attributes other than
ism:classification
            with a value of [U] and ism:ownerProducer with a value of [USA],
then
            there must exist two independent HandlingAssertion elements, one
with
            attribute scope specified with a value of [TDO] and the other
with
            attribute scope specified with a value of [PAYL].

            Human Readable: If the single IC-EDH contains classified
information
            within a ResponsibleEntity, then the instance is not the minimal
case
            and must specify markings for the payload and the TDO instance
```

```
separately.
        </sch:assert>
                </sch:rule>
              </sch:pattern>
```

2.24 - .//Rules/IC-TDF_ID_00024.sch

Rule Description: IC-TDF-ID-00024 [IC-TDF-ID-00024][Error] For element TrustedDataObject, if there is a single HandlingAssertion element and it contains any ntk:Access or ntk:ExternalAccess elements which specify any ISM attributes other than ism:classification with a value of [U] and ism:ownerProducer with a value of [USA], then there must exist two independent HandlingAssertion elements, one with attribute scope specified with a value of [TDO] and the other with attribute scope specified with a value of [PAYL]. Human Readable: If the single IC-EDH contains classified information within NTK information, then the instance is not the minimal case and must specify markings for the payload and the TDO instance separately.

Code Description: For TrustedDataObject elements with a single HandlingAssertion descendant element and NTK elements which specify ISM attributes other than ism:classification with a value of [U] and ism:ownerProducer with a value of [USA], we ensure that there exists two independent HandlingAssertion elements, one of scope [TDO] and one of [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="IC-TDF-ID-00024">


                     <sch:rule context="tdf:TrustedDataObject   [count(.//
tdf:HandlingAssertion)=1]   [some $ntk in tdf:HandlingAssertion//ntk:*
satisfies    ($ntk/@ism:classification    and (not($ntk/
@ism:classification='U') or not($ntk/@ism:ownerProducer='USA')))]">
                           <sch:assert
test="not(tdf:HandlingAssertion[@tdf:scope='TDO PAYL'])" flag="error">
            [IC-TDF-ID-00024][Error] For element TrustedDataObject, if there
is
            a single HandlingAssertion element and it contains any ntk:Access
or
            ntk:ExternalAccess lements which specify any ISM attributes other
than
            ism:classification with a value of [U] and ism:ownerProducer with
a
            value of [USA], then there must exist two independent
HandlingAssertion
            elements, one with attribute scope specified with a value of
[TDO] and
            the other with attribute scope specified with a value of [PAYL].

            Human Readable: If the single IC-EDH contains classified
information
            within NTK information, then the instance is not the minimal case
            and must specify markings for the payload and the TDO instance
separately.
        </sch:assert>
```

```
                </sch:rule>
            </sch:pattern>
```

2.25 - .//Rules/appliesToState/IC-TDF_ID_00025.sch

Rule Description: IC-TDF-ID-00025 [IC-TDF-ID-00025][Error] Attribute @appliesToState is only allowed when TDO payload attrbute @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, we ensure that there is a payload element with attribute isEncrpyted 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="IC-TDF-ID-00025">


                     <sch:rule
context="tdf:TrustedDataObject[tdf:HandlingAssertion/@tdf:appliesToState]">
                           <sch:assert test="./*/@tdf:isEncrypted = true()"
flag="error">
            [IC-TDF-ID-00024][Error] Attribute @appliesToState is only
allowed when
            TDO payload attrbute @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>
```

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

Rule Description: IC-TDF-ID-00026 [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, we ensure 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 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">
            [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>
```

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

Rule Description: IC-TDF-ID-00027 [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, we ensure 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 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">
            [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>
```

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

Rule Description: IC-TDF-ID-00028 [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:

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
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">
            [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>
```

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

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

Code Description:

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 context="tdf:TrustedDataObject/tdf:Assertion/
tdf:*[@tdf:isEncrypted=true()]">
                           <sch:assert test="count(parent::node()/
tdf:StatementMetadata[@tdf:appliesToState='unencrypted']/edh:ExternalEdh)= 1"
                                 flag="error">


            [IC-TDF-ID-00030][Error] If statement attribute
@isEncrypted="true", the statement metadata
            that contains @appliesToState=”unencrypted”  must
            contain an edh:externalEDH

            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>
```

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

Rule Description: IC-TDF-ID-00031 [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 it's encrypted state, and unencrypted state.

Code Description:

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 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">
            [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
            it's encrypted state, and unencrypted state.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

2.31 - .//Rules/appliesToState/IC-TDF_ID_00032.sch

Rule Description: IC-TDF-ID-00032 [IC-TDF-ID-00032][Error] Attribute @appliesToState is only allowed when TDO statement attrbute @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, we ensure that there is a statement element with attribute isEncrpyted 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="IC-TDF-ID-00032">


                     <sch:rule context="tdf:TrustedDataObject/
tdf:Assertion[tdf:StatementMetadata/@tdf:appliesToState]">
                           <sch:assert test="./*/@tdf:isEncrypted = true()"
flag="error">
            [IC-TDF-ID-00032][Error] Attribute @appliesToState is only
allowed when
            TDO statement attrbute @isEncrypted equals "true".

            Human Readable:  StatementMetadata state applicability can only
be defined
            when an encrypted statement is present.
        </sch:assert>
                  </sch:rule>
               </sch:pattern>
```

Chapter 3 - Abstract Patterns

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

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

Rule Description: This is the root file for the IC-TDF Schematron rule set. It loads all of the required CVEs, declares some global variables, and includes all of the Rule .sch files.

Code Description: This is the root file for the IC-TDF Schematron rule set. It loads all of the required CVEs, declares some global variables, and includes all of the Rule .sch files.

Schematron Code:

```
<?ICEA master?><!-- UNCLASSIFIED --><!-- Notices - Distribution Notice:
            This document is being made available by the Intelligence
Community Chief Information Officer
            to Federal, State, Local, Tribal, and Foreign Partners and
associated contractors. Approval for
            any further distribution must be coordinated via the Intelligence
Community Chief Information
            Officer, Mission Engagement Division at standardssupport@dni.gov--
><!-- WARNING:
    Once compiled into an XSLT the result will
    be the aggregate classification of all the CVES
    and included .sch files
-->
<sch:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                           xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                           queryBinding="xslt2">
                  <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="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 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 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 ID Rules -->
<!--****************************-->

<!--(U) appliesToState-->
   <sch:include href="./Rules/appliesToState/IC-TDF_ID_00025.sch"/>
                <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"/>
                <sch:include href="./Rules/appliesToState/IC-
TDF_ID_00032.sch"/>

                <!--(U) -->
   <sch:include href="./Rules/IC-TDF_ID_00001.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00002.sch"/>
                <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_00006.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00007.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00008.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00009.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00010.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00011.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00012.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00013.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00014.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00015.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_00020.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00021.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00022.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00023.sch"/>
                <sch:include href="./Rules/IC-TDF_ID_00024.sch"/>
```

```
    </sch:schema>
    <!-- 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.