



Guide to Schematron Rules and Patterns

UIAS 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 - Schematron Phase VALUECHECK	3
Chapter 3 - Schematron Phase NON-ENV	4
Chapter 4 - Rules	5
4.1 - ../Rules/UIAS_ID_00001.sch	6
4.2 - ../Rules/UIAS_ID_00004.sch	7
4.3 - ../Rules/UIAS_ID_00005.sch	8
4.4 - ../Rules/UIAS_ID_00006.sch	9
4.5 - ../Rules/UIAS_ID_00007.sch	10
4.6 - ../Rules/UIAS_ID_00008.sch	11
4.7 - ../Rules/UIAS_ID_00009.sch	12
4.8 - ../Rules/UIAS_ID_00011.sch	13
4.9 - ../Rules/UIAS_ID_00012.sch	14
4.10 - ../Rules/UIAS_ID_00014.sch	15
4.11 - ../Rules/UIAS_ID_00016.sch	16
4.12 - ../Rules/UIAS_ID_00019.sch	17
4.13 - ../Rules/UIAS_ID_00021.sch	18
4.14 - ../Rules/UIAS_ID_00022.sch	19
4.15 - ../Rules/UIAS_ID_00023.sch	20
4.16 - ../Rules/UIAS_ID_00024.sch	21
4.17 - ../Rules/UIAS_ID_00025.sch	22
4.18 - ../Rules/UIAS_ID_00026.sch	23
4.19 - ../Rules/UIAS_ID_00028.sch	24
4.20 - ../Rules/UIAS_ID_00030.sch	25
4.21 - ../Rules/UIAS_ID_00036.sch	26
4.22 - ../Rules/UIAS_ID_00047.sch	27
4.23 - ../Rules/UIAS_ID_00050.sch	28
4.24 - ../Rules/UIAS_ID_00051.sch	29
4.25 - ../Rules/UIAS_ID_00052.sch	30
4.26 - ../Rules/UIAS_ID_00053.sch	31
4.27 - ../Rules/UIAS_ID_00056.sch	32
4.28 - ../Rules/UIAS_ID_00057.sch	33
4.29 - ../Rules/UIAS_ID_00065.sch	34
4.30 - ../Rules/UIAS_ID_00066.sch	35
4.31 - ../Rules/UIAS_ID_00067.sch	36
4.32 - ../Rules/UIAS_ID_00068.sch	37
4.33 - ../Rules/UIAS_ID_00069.sch	38
4.34 - ../Rules/UIAS_ID_00070.sch	39
4.35 - ../Rules/UIAS_ID_00071.sch	40
4.36 - ../Rules/UIAS_ID_00072.sch	41
4.37 - ../Rules/UIAS_ID_00073.sch	42
4.38 - ../Rules/UIAS_ID_00074.sch	43

4.39 - //Rules/UIAS_ID_00075.sch	44
4.40 - //Rules/UIAS_ID_00076.sch	45
4.41 - //Rules/UIAS_ID_00077.sch	46
4.42 - //Rules/UIAS_ID_00078.sch	47
4.43 - //Rules/UIAS_ID_00079.sch	48
4.44 - //Rules/UIAS_ID_00080.sch	49
4.45 - //Rules/UIAS_ID_00081.sch	50
Chapter 5 - Abstract Patterns	51
5.1 - //Lib/ValidateEmbeddedValueExistenceInList.sch	52
5.2 - //Lib/ValidateTokenValuesExistInNamespaceList.sch	54
5.3 - //Lib/ValidateTokenValuesExistenceInList.sch	55
5.4 - //Lib/ValidateValidationEnvCVE.sch	56
5.5 - //Lib/ValidateValidationEnvSchema.sch	57
5.6 - //Lib/ValidateValueExistenceInList.sch	58
Chapter 6 - Schematron Schema	59
6.1 - //UIAS_XML.sch	60
Chapter 7 - Removed Rules	65
7.1 - //Rules/deleted/UIAS_ID_00002.sch	65
7.2 - //Rules/deleted/UIAS_ID_00003.sch	65
7.3 - //Rules/deleted/UIAS_ID_00010.sch	65
7.4 - //Rules/deleted/UIAS_ID_00013.sch	65
7.5 - //Rules/deleted/UIAS_ID_00015.sch	65
7.6 - //Rules/deleted/UIAS_ID_00017.sch	65
7.7 - //Rules/deleted/UIAS_ID_00018.sch	65
7.8 - //Rules/deleted/UIAS_ID_00020.sch	66
7.9 - //Rules/deleted/UIAS_ID_00027.sch	66
7.10 - //Rules/deleted/UIAS_ID_00029.sch	66
7.11 - //Rules/deleted/UIAS_ID_00031.sch	66
7.12 - //Rules/deleted/UIAS_ID_00032.sch	66
7.13 - //Rules/deleted/UIAS_ID_00033.sch	66
7.14 - //Rules/deleted/UIAS_ID_00034.sch	66
7.15 - //Rules/deleted/UIAS_ID_00035.sch	67
7.16 - //Rules/deleted/UIAS_ID_00037.sch	67
7.17 - //Rules/deleted/UIAS_ID_00038.sch	67
7.18 - //Rules/deleted/UIAS_ID_00039.sch	67
7.19 - //Rules/deleted/UIAS_ID_00040.sch	67
7.20 - //Rules/deleted/UIAS_ID_00041.sch	67
7.21 - //Rules/deleted/UIAS_ID_00042.sch	67
7.22 - //Rules/deleted/UIAS_ID_00043.sch	68
7.23 - //Rules/deleted/UIAS_ID_00044.sch	68
7.24 - //Rules/deleted/UIAS_ID_00045.sch	68
7.25 - //Rules/deleted/UIAS_ID_00046.sch	68
7.26 - //Rules/deleted/UIAS_ID_00048.sch	68
7.27 - //Rules/deleted/UIAS_ID_00049.sch	68
7.28 - //Rules/deleted/UIAS_ID_00054.sch	68
7.29 - //Rules/deleted/UIAS_ID_00055.sch	69
7.30 - //Rules/deleted/UIAS_ID_00058.sch	69
7.31 - //Rules/deleted/UIAS_ID_00059.sch	69
7.32 - //Rules/deleted/UIAS_ID_00060.sch	69

7.33 - ../Rules/deleted/UIAS_ID_00061.sch	69
7.34 - ../Rules/deleted/UIAS_ID_00062.sch	69
7.35 - ../Rules/deleted/UIAS_ID_00063.sch	69
7.36 - ../Rules/deleted/UIAS_ID_00064.sch	70

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for UIAS. This guide is generated from the UIAS Schematron rules and provides a consolidated reference for the business rules of this specification.

1.2 - Overview

Chapters 2 through 3 are listings of each Phase defined for UIAS. There is a chapter for each Phase listing the rule number and rule description for each rule in the Phase. The rule number is a link to the full code listing for that rule.

Chapter 4 is a listing of all the numbered rules in UIAS. For each rule, there is a rule description, a code description, and a code block with the Schematron rule.

Chapter 5 is a listing of abstract patterns used in UIAS. The abstract patterns may be used in numbered rules or provided as reference for use in rules developed by users of UIAS. Each abstract pattern has a code description and a code block with the abstract Schematron pattern.

Chapter 6 is a listing of the master UIAS 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 7 is a listing of rules that have been deleted.

1.3 - Schematron

The business rules for UIAS are encoded using ISO Schematron. Schematron is a rule-based validation language that uses XML Path Language to make assertions about an XML document.

UIAS 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 UIAS's Schematron rules.

Chapter 2 - Schematron Phase VALUECHECK

Rules in Phase

Schematron Phase VALUECHECK	
Rule Id	Rule
UIAS-ID-00081	[Error] The SAR tokens in fineAccessControls must start with a substring after SAR- and before : that exists in the SAR Source Authorities CVE.

Chapter 3 - Schematron Phase NON-ENV

Rules in Phase

Schematron Phase NON-ENV	
Rule Id	Rule
UIAS-ID-00081	[Error] The SAR tokens in fineAccessControls must start with a substring after SAR- and before : that exists in the SAR Source Authorities CVE.

Chapter 4 - Rules

All of the numbered Rules for UIAS 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 UIAS-ID-XXXXX, with rule files named UIAS_ID_XXXXX.sch. Any other heading indicates a supporting file that may influence a rule but is not actually a numbered rule.

4.1 - ../Rules/UIAS_ID_00001.sch

Rule Description

[UIAS-ID-00001][Error] adminOrganization must be present. Human Readable: The element with @Name=adminOrganization must be present.

Code Description

check for presence of adminOrganization

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="UIAS-ID-00001">
    <sch:rule id="UIAS-ID-00001-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='adminOrganization']"
                    flag="error"
                    role="error">[UIAS-ID-00001][Error] The element with @Name=adminOrganization must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.2 - ../Rules/UIAS_ID_00004.sch

Rule Description

[UIAS-ID-00004][Error] If ATOSStatus is present, entityType must be a Non-Person Entity. Human Readable: If element with @Name=ATOSStatus is present, then element with @Name=entityType must have one of the values in CVEEnumUIASNonPersonEntityType.

Code Description

ATOSStatus must contain a value if the entity is NPE.

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="UIAS-ID-00004">
  <sch:rule id="UIAS-ID-00004-R1"
    context="saml:AttributeStatement[saml:Attribute[@Name='ATOSStatus']]/saml:Attribute[@Name='entityType']">
    <sch:assert test="saml:AttributeValue[some $entityType in $nonPersonEntityList satisfies $entityType = normalize-space(.)]"
      flag="error"
      role="error">[UIAS-ID-00004][Error] If element with @Name=ATOSStatus is present, then element with @Name=entityType must have one of the values in
CVEEnumUIASNonPersonEntityType.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

4.3 - ../Rules/UIAS_ID_00005.sch

Rule Description

[UIAS-ID-00005][Error] If a saml attribute with @Name=aICP is true, then the saml attribute with @Name=isICMember must be true. Human Readable: If aICP is true, then isICmember must be true.

Code Description

For a saml attribute whose name is isICMember, verify that if there is a saml attribute with name aICP and its value is true, then the value of isICMember must be 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="UIAS-ID-00005">
    <sch:rule id="UIAS-ID-00005-R1" context="saml:Attribute[@Name='isICMember']">
        <sch:assert test="if (normalize-space(string(..saml:Attribute[@Name='aICP']/saml:AttributeValue)) = 'true') then saml:AttributeValue['true' = normalize-space(.)] else
true()"
                                flag="error"
                                role="error">[UIAS-ID-00005][Error] If aICP is true, then isICmember must be true.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.4 - ../Rules/UIAS_ID_00006.sch

Rule Description

[UIAS-ID-00006][Error] If aICP is present, then entityType must be a member of CVEnumUIASPersonEntityType. Human Readable: If an element with @Name=aICP has a value, then an element with @Name=entityType must have one of the values in CVEnumUIASPersonEntityType.

Code Description

Check for presence of attribute aICP, if present check that attribute entityType is a PE.

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="UIAS-ID-00006">
    <sch:rule id="UIAS-ID-00006-R1"
        context="saml:AttributeStatement[saml:Attribute[@Name='aICP']]/saml:Attribute[@Name='entityType']">
        <sch:assert test="saml:AttributeValue[some $entityType in $personEntityList satisfies $entityType = tokenize(normalize-space(string(.)), ' ')[1]]"
            flag="error"
            role="error">[UIAS-ID-00006][Error] If an element with @Name=aICP has a value, then an element with @Name=entityType must have one of the values in
CVEnumUIASPersonEntityType.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.5 - ../Rules/UIAS_ID_00007.sch

Rule Description

[UIAS-ID-00007][Error] If the element with @Name-Role uses C2S namespace, the scope value must exist in the list of allowed C2S scope values.

Code Description

If element with @Name-Role uses C2S namespace, invoke abstract rule ValueExistsInList to check if the value exists in the list of allowed C2S scope values.

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="UIAS-ID-00007" is-a="ValidateTokenValuesExistInNamespaceList">
    <sch:param name="context" value="saml:Attribute[@Name='role']"/>
    <sch:param name="namespaceValue" value="saml:AttributeValue[1]"/>
    <sch:param name="namespaceUsed" value="'C2S'"/>
    <sch:param name="tokenList" value="$roleC2SScopeList"/>
    <sch:param name="errMsg"
        value="'[UIAS-ID-00007][Error] If the element with @Name-Role uses C2S namespace, the scope value must exist in the list of allowed C2S scope values.'"/>
</sch:pattern>
```

4.6 - ../Rules/UIAS_ID_00008.sch

Rule Description

[UIAS-ID-00008][Error] Clearance must be present. Human Readable: An element with @Name=clearance must be present.

Code Description

Clearance must be present.

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="UIAS-ID-00008">
    <sch:rule id="UIAS-ID-00008-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='clearance']" flag="error" role="error">[UIAS-ID-00008][Error] An element with @Name=clearance must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```


4.7 - ../Rules/UIAS_ID_00009.sch

Rule Description

[UIAS-ID-00009][Error] clearance must contain at least one value. Human Readable: An element with @Name=clearance must have at least one value.

Code Description

clearance must contain one 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="UIAS-ID-00009">
    <sch:rule id="UIAS-ID-00009-R1" context="saml:Attribute[@Name='clearance']">
        <sch:assert test="count(tokenize(normalize-space(string(saml:AttributeValue[1])), ' ')) > 0"
            flag="error"
            role="error">[UIAS-ID-00009][Error] An element with @Name=clearance must have at least one value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.8 - ../Rules/UIAS_ID_00011.sch

Rule Description

[UIAS-ID-00011][Error] countryOfAffiliation must be present. Human Readable: An element with @Name=countryOfAffiliation must be present.

Code Description

countryOfAffiliation must be present.

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="UIAS-ID-00011">
    <sch:rule id="UIAS-ID-00011-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='countryOfAffiliation']"
                    flag="error"
                    role="error">[UIAS-ID-00011][Error] An element with @Name=countryOfAffiliation must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.9 - ../Rules/UIAS_ID_00012.sch

Rule Description

[UIAS-ID-00012][Error] countryOfAffiliation must contain at least one value. Human Readable: An element with @Name=countryOfAffiliation must contain at least one value.

Code Description

countryOfAffiliation must contain at least one 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="UIAS-ID-00012">
  <sch:rule id="UIAS-ID-00012-R1"
    context="saml:Attribute[@Name='countryOfAffiliation']">
    <sch:assert test="count(tokenize(normalize-space(string(saml:AttributeValue[1])), ' ')) > 0"
      flag="error"
      role="error">[UIAS-ID-00012][Error] An element with @Name=countryOfAffiliation must contain at least one value.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

4.10 - ../Rules/UIAS_ID_00014.sch

Rule Description

[UIAS-ID-00014][Error] digitalIdentifier must be present. Human Readable: An element with @Name=digitalIdentifier must be present.

Code Description

digitalIdentifier must be present.

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="UIAS-ID-00014">
    <sch:rule id="UIAS-ID-00014-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='digitalIdentifier']"
                    flag="error"
                    role="error">[UIAS-ID-00014][Error] An element with @Name=digitalIdentifier must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.11 - ../Rules/UIAS_ID_00016.sch

Rule Description

[UIAS-ID-00016][Error] dutyOrganization must be present. Human Readable: An element with @Name=dutyOrganization must be present.

Code Description

dutyOrganization must be present.

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="UIAS-ID-00016">
    <sch:rule id="UIAS-ID-00016-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='dutyOrganization']"
                    flag="error"
                    role="error">[UIAS-ID-00016][Error] An element with @Name=dutyOrganization must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.12 - ../Rules/UIAS_ID_00019.sch

Rule Description

[UIAS-ID-00019][Error] entityType must be present. Human Readable: An element with @Name=entityType must be present.

Code Description

entityType must be present.

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="UIAS-ID-00019">
    <sch:rule id="UIAS-ID-00019-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='entityType']" flag="error" role="error">[UIAS-ID-00019][Error] An element with @Name=entityType must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.13 - ../Rules/UIAS_ID_00021.sch

Rule Description

[UIAS-ID-00021][Error] entityType value must be a member of the UIAS internal CEnumUIASEntityType. Human Readable: The element with @Name=entityType must be a member of the UIAS internal CEnum CEnumUIASEntityType.

Code Description

check that entityType value is a member of the UIAS internal CEnum CEnumUIASEntityType.

Schematron Code

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

<sch:pattern is-a="ValidateTokenValuesExistenceInList" id="UIAS-ID-00021">
    <sch:param name="context" value="saml:Attribute[@Name='entityType']"/>
    <sch:param name="list" value="$entityTypeList"/>
    <sch:param name="searchTermList" value="normalize-space(saml:AttributeValue)"/>
    <sch:param name="errMsg"
        value="' [UIAS-ID-00021][Error] The element with @Name=entityType must be a member of the UIAS internal CEnum CEnumUIASEntityType. '"/>
</sch:pattern>
```

4.14 - ../Rules/UIAS_ID_00022.sch

Rule Description

[UIAS-ID-00022][Error] If entityType is NPE, then ATOSStatus must be present. Human Readable: If element with @Name=entityType has a value of one of the values in CVEnumUIASNonPersonEntityType, then element with @Name=ATOSStatus must have a value.

Code Description

Check if attribute entityType = NPE, if true make sure there is an ATOSStatus 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="UIAS-ID-00022">
    <sch:rule id="UIAS-ID-00022-R1" context="saml:Attribute[@Name='entityType']">
        <sch:assert test="( if (saml:AttributeValue[some $entityType in $nonPersonEntityList satisfies $entityType = normalize-space(.)]) then (count(tokenize(normalize-
space(string(..saml:Attribute[@Name='ATOSStatus']/saml:AttributeValue)), ' ')) > 0) else true())"
            flag="error"
            role="error">[UIAS-ID-00022][Error] If element with @Name=entityType has a value of one of the values in CVEnumUIASNonPersonEntityType, then element with
@Name=ATOSStatus must have a value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```


4.15 - ../Rules/UIAS_ID_00023.sch

Rule Description

[UIAS-ID-00023][Error] If entityType is NPE, then lifeCycleStatus must be present. Human Readable: If element with @Name=entityType has a value of one of the values in CVEnumUIASNonPersonEntityType, then element with @Name=lifeCycleStatus must have a value.

Code Description

Check if attribute entityType = NPE, if true make sure there is a lifeCycleStatus 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="UIAS-ID-00023">
    <sch:rule id="UIAS-ID-00023-R1" context="saml:Attribute[@Name='entityType']">
        <sch:assert test="( if (saml:AttributeValue[some $entityType in $nonPersonEntityList satisfies $entityType = normalize-space(.)]) then (count(tokenize(normalize-
space(string(..saml:Attribute[@Name='lifeCycleStatus']/saml:AttributeValue)), ' ')) > 0) else true())"
            flag="error"
            role="error">[UIAS-ID-00023][Error] If element with @Name=entityType has a value of one of the values in CVEnumUIASNonPersonEntityType, then element with
@Name=lifeCycleStatus must have a value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.16 - ../Rules/UIAS_ID_00024.sch

Rule Description

[UIAS-ID-00024][Error] If entityType is PE, then aICP must be present. Human Readable: If element with @Name=entityType has a value of one of the values in CVEnumUIASPersonEntityType, then element with @Name=aICP must have a value.

Code Description

Check for attribute entityType = PE, if true make sure there is an aICP attribute present.

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="UIAS-ID-00024">
    <sch:rule id="UIAS-ID-00024-R1" context="saml:Attribute[@Name='entityType']">
        <sch:assert test="( if (saml:AttributeValue[some $entityType in $personEntityList satisfies $entityType = normalize-space(.)]) then (count(tokenize(normalize-
space(string(..saml:Attribute[@Name='aICP']/saml:AttributeValue)), ' ')) > 0) else true())"
            flag="error"
            role="error">[UIAS-ID-00024][Error] If element with @Name=entityType has a value of one of the values in CVEnumUIASPersonEntityType, then element with
@Name=aICP must have a value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.17 - ../Rules/UIAS_ID_00025.sch

Rule Description

[UIAS-ID-00025][Error] fineAccessControls must be present. Human Readable: An element with @Name=fineAccessControls must be present.

Code Description

fineAccessControls must be present.

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="UIAS-ID-00025">
    <sch:rule id="UIAS-ID-00025-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='fineAccessControls']"
                    flag="error"
                    role="error">[UIAS-ID-00025][Error] An element with @Name=fineAccessControls must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.18 - ../Rules/UIAS_ID_00026.sch

Rule Description

[UIAS-ID-00026][Error] fineAccessControls must contain at least one value. Human Readable: An element with @Name=fineAccessControls must contain at least one value.

Code Description

fineAccessControls must contain one 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="UIAS-ID-00026">
  <sch:rule id="UIAS-ID-00026-R1"
    context="saml:Attribute[@Name='fineAccessControls']">
    <sch:assert test="count(tokenize(normalize-space(string(saml:AttributeValue[1])), ' ')) > 0"
      flag="error"
      role="error">[UIAS-ID-00026][Error] An element with @Name=fineAccessControls must contain at least one value.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

4.19 - ../Rules/UIAS_ID_00028.sch

Rule Description

[UIAS-ID-00028][Error] isICMember must be present. Human Readable: An element with @Name=isICMember must be present.

Code Description

isICMember must be present.

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="UIAS-ID-00028">
    <sch:rule id="UIAS-ID-00028-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='isICMember']" flag="error" role="error">[UIAS-ID-00028][Error] An element with @Name=isICMember must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.20 - ../Rules/UIAS_ID_00030.sch

Rule Description

[UIAS-ID-00030][Error] If lifeCycleStatus is present, entityType must be a Non-Person Entity. Human Readable: If element with @Name=lifeCycleStatus is present, then element with @Name=entityType must have a value from CVEnumUIASNonPersonEntityType.

Code Description

If lifeCycleStatus is present, entityType must be a member of CVEnumUIASNonPersonEntityType.

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="UIAS-ID-00030">
  <sch:rule id="UIAS-ID-00030-R1"
    context="saml:AttributeStatement[saml:Attribute[@Name='lifeCycleStatus']]/saml:Attribute[@Name='entityType']">
    <sch:assert test="saml:AttributeValue[some $entityType in $nonPersonEntityList satisfies $entityType = normalize-space(.)]"
      flag="error"
      role="error">[UIAS-ID-00030][Error] If element with @Name=lifeCycleStatus is present, then element with @Name=entityType must have a value from
CVEnumUIASNonPersonEntityType.</sch:assert>
    </sch:rule>
  </sch:pattern>
```

4.21 - ../Rules/UIAS_ID_00036.sch

Rule Description

[UIAS-ID-00036][Error] The value of dutyOrganizationUnit element must begin with the value of the dutyOrganization element. Human Readable: The value of element with @Name=dutyOrganizationUnit must begin with the value of the element with @Name=dutyOrganization.

Code Description

The value of dutyOrganizationUnit must begin with the value of dutyOrganization.

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="UIAS-ID-00036">
    <sch:rule id="UIAS-ID-00036-R1"
        context="saml:Attribute[@Name='dutyOrganizationUnit']">
        <sch:assert test="normalize-space(string(..//saml:Attribute[@Name='dutyOrganization'])[1]/saml:AttributeValue)) = tokenize(normalize-
space(string(saml:AttributeValue[1])),':')[1]"
            flag="error"
            role="error">[UIAS-ID-00036][Error] The value of element with @Name=dutyOrganizationUnit must begin with the value of the element with
@Name=dutyOrganization.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.22 - ../Rules/UIAS_ID_00047.sch

Rule Description

[UIAS-ID-00047][Error] auditRoutingOrganization must be present. Human Readable: The element with @Name=auditRoutingOrganization must be present.

Code Description

check for presence of auditRoutingOrganization

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="UIAS-ID-00047">
    <sch:rule id="UIAS-ID-00047-R1" context="saml:AttributeStatement">
        <sch:assert test="saml:Attribute[@Name='auditRoutingOrganization']"
                    flag="error"
                    role="error">[UIAS-ID-00047][Error] The element with @Name=auditRoutingOrganization must be present.</sch:assert>
    </sch:rule>
</sch:pattern>
```


4.23 - ../Rules/UIAS_ID_00050.sch

Rule Description

[UIAS-ID-00050][Error] If the element with @Name-Role uses PAAS namespace, the scope value must exist in the list of allowed PAAS scope values.

Code Description

If element with @Name-Role uses PAAS namespace, invoke abstract rule ValueExistsInList to check if the value exists in the list of allowed PAAS scope values.

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="UIAS-ID-00050" is-a="ValidateTokenValuesExistInNamespaceList">
    <sch:param name="context" value="saml:Attribute[@Name='role']"/>
    <sch:param name="namespaceValue" value="saml:AttributeValue[1]"/>
    <sch:param name="namespaceUsed" value="'PAAS'"/>
    <sch:param name="tokenList" value="$rolePAASScopeList"/>
    <sch:param name="errMsg"
        value="'[UIAS-ID-00050][Error] If the element with @Name-Role uses PAAS namespace, the scope value must exist in the list of allowed PAAS scope values.'"/>
</sch:pattern>
```

4.24 - ../Rules/UIAS_ID_00051.sch

Rule Description

[UIAS-ID-00051][Error] There can only be one saml:Attribute with any given @Name

Code Description

Use the context to look for prior siblings with the same name. Fail if any are found.

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="UIAS-ID-00051">
    <sch:rule id="UIAS-ID-00051-R1"
        context="saml:Attribute[@Name=preceding-sibling::saml:Attribute/@Name]">
        <sch:assert test="false()" flag="error" role="error">[UIAS-ID-00051][Error] Duplicate saml:Attribute/@Name are not allowed. More than one of "
    <sch:value-of select="../@Name"/>" was found.
    </sch:assert>
    </sch:rule>
</sch:pattern>
```

4.25 - ../Rules/UIAS_ID_00052.sch

Rule Description

[UIAS-ID-00052][Error] There can only be one and only one saml:AttributeValue in any saml:Attribute

Code Description

Use the context to look for prior siblings with the same name. Fail if any are found.

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="UIAS-ID-00052">
    <sch:rule id="UIAS-ID-00052-R1"
        context="saml:Attribute[count(saml:AttributeValue)!=1]">
        <sch:assert test="not(count(saml:AttributeValue)=0)" flag="error" role="error">[UIAS-ID-00052][Error] A saml:AttributeValue must be specified there was not one found.</
sch:assert>
        <sch:assert test="not(count(saml:AttributeValue)>1)" flag="error" role="error">[UIAS-ID-00052][Error] More than one saml:AttributeValue must not be be specified
there were
        <sch:value-of select="count(saml:AttributeValue)"/>found.
    </sch:assert>
    </sch:rule>
</sch:pattern>
```

4.26 - ../Rules/UIAS_ID_00053.sch

Rule Description

[UIAS-ID-00053][Error] The RoleOrg value of the C2S & PAAS role taxonomy appended with the prefix "USA." MUST be a value found in CVEnumUSAgencyAcronym.xml.

Code Description

Tokenize on space to get all the roles for each role tokenize on - to get the parts, verify the 1st part is C2S or PAAS if so verify the 2nd part appended with "USA." prefix is in USAgency.

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="UIAS-ID-00053">
    <sch:rule id="UIAS-ID-00053-R1" context="saml:Attribute[@Name = 'role']">
        <sch:assert test="every $token in tokenize(normalize-space(string(saml:AttributeValue[1])), ' ') satisfies (not(tokenize(string($token), '-') [1]='C2S') and
not(tokenize(string($token), '-') [1]='PAAS') ) or (every $C2SorPAAS token in tokenize(string($token), '-') [2] satisfies concat('USA.', $C2SorPAAS token) = $usAgencyList or (some $item in
$usAgencyList satisfies (matches(normalize-space(concat('USA.', $C2SorPAAS token)), concat('^',$item,'$')))))"
            flag="error"
            role="error">[UIAS-ID-00053][Error] The RoleOrg value of the C2S & PAAS role taxonomy appended with the prefix "USA." MUST be a value found in
CVEnumUSAgencyAcronym.xml.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.27 - ../Rules/UIAS_ID_00056.sch

Rule Description

[UIAS-ID-00056][Warning] The saml:Attribute/@Name SHOULD be in the CVE CVEEnumUIASSchemaTypes.xml unless it is a local/extended attribute.

Code Description

Given the AttributeName, lookup that name in the SchemaTypes to make sure it's a valid AttributeName

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="UIAS-ID-00056">
    <sch:rule id="UIAS-ID-00056-R1" context="saml:Attribute">
        <sch:let name="attributeName" value="./@Name"/>
        <sch:assert test="$attributeName = $schemaTypeList//cve:Value or (some $item in $schemaTypeList//cve:Value satisfies (matches(normalize-space($attributeName),
concat('^',$item,'$'))))"
                                flag="error"
                                role="error">[UIAS-ID-00056][Warning] The saml:Attribute/@Name SHOULD be in the CVE CVEEnumUIASSchemaTypes.xml
        <sch:value-of select="$attributeName"/>was not in CVEEnumUIASSchemaTypes it MAY be a local/extended attribute.
    </sch:assert>
    </sch:rule>
</sch:pattern>
```

4.28 - ../Rules/UIAS_ID_00057.sch

Rule Description

[UIAS-ID-00057][Error] The xsi:type of a UIAS attribute MUST match the description for that attribute Name in CVEnumUIASSchemaTypes.xml

Code Description

Determine the xsi:type of the AttributeValue and check against the correct xsi:type for that AttributeName in the schema

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="UIAS-ID-00057">
    <sch:rule id="UIAS-ID-00057-R1" context="saml:Attribute">
        <sch:let name="attributeName" value="./@Name"/>
        <sch:let name="type" value="./saml:AttributeValue[1]/@xsi:type"/>
        <sch:assert test="some $item in $schemaTypeList satisfies ( (matches(normalize-space($attributeName), concat('^',$item//cve:Value,'$')) and (matches(normalize-
space($type), concat('^',$item//cve:Description,'$')) ) or not(some $item in $schemaTypeList satisfies ( (matches(normalize-space($attributeName), concat('^',$item//cve:Value,'$')) ) )"
            flag="error"
            role="error">[UIAS-ID-00057][Error] The saml:Attribute/@Name MUST be in the CVE CVEnumUIASSchemaTypes.xml associated with the description matching the
xsi:type</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.29 - ../Rules/UIAS_ID_00065.sch

Rule Description

[UIAS-ID-00065][Error] The saml:Attribute[@Name='countryOfAffiliation']/saml:AttributeValue element cannot have "NATO" as a value. Human Readable: The SAML AttributeValue element of a SAML Attribute element with attribute Name='countryOfAffiliation' cannot have NATO as a 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="UIAS-ID-00065">
    <sch:rule id="UIAS-ID-00065-R1"
        context="saml:Attribute[@Name='countryOfAffiliation']/saml:AttributeValue">
        <sch:assert test="not(contains(normalize-space(.),'NATO'))"
            flag="error"
            role="error">[UIAS-ID-00065][Error] The saml:Attribute[@Name='countryOfAffiliation']/saml:AttributeValue element cannot have "NATO" as a value. Human
Readable: The SAML AttributeValue element of a SAML Attribute element with attribute Name='countryOfAffiliation' cannot have NATO as a value.</sch:assert>
        </sch:rule>
    </sch:pattern>
```

4.30 - ../Rules/UIAS_ID_00066.sch

Rule Description

[UIAS-ID-00066][Error] If the xsi:type defined is in the CVENumUIASSchemaType CVE, then the FriendlyName must be the concatenated string of the UIAS URN ('urn:us:gov:ic:uias:') and the Name.

Code Description

If @xsi:type is a type defined in the CVENumUIASSchemaType CVE, then @FriendlyName must be the concatenated string of the UIAS URN ('urn:us:gov:ic:uias:') and @Name.

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="UIAS-ID-00066">
  <sch:rule id="UIAS-ID-00066-R1" context="saml:Attribute">
    <sch:let name="attributeName" value="./@Name"/>
    <sch:let name="uiasURN" value="'urn:us:gov:ic:uias:'"/>
    <sch:let name="attributeFriendlyName" value="./@FriendlyName"/>
    <sch:let name="type" value="./saml:AttributeValue[1]/@xsi:type"/>
    <sch:assert test="if (some $item in $schemaTypeList satisfies (matches(normalize-space($type), concat('^',$item//cve:Description,'$')))) then
not(compare($attributeFriendlyName, concat($uiasURN, $attributeName))) else true()"
      flag="error"
      role="error">[UIAS-ID-00066][Error] If the xsi:type defined is in the CVENumUIASSchemaType CVE, then the FriendlyName must be the concatenated string of the
UIAS URN ('urn:us:gov:ic:uias:') and the Name.</sch:assert>
  </sch:rule>
</sch:pattern>
```


4.31 - ./Rules/UIAS_ID_00067.sch

Rule Description

[UIAS-ID-00067][Warning] uias:DESVersion 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="UIAS-ID-00067">
    <sch:rule id="UIAS-ID-00067-R1" context="*[@uias:DESVersion]">
        <sch:assert test="matches(@uias:DESVersion, '^202111(-.{1,23})?$' )"
            flag="warning"
            role="warning">[UIAS-ID-00067][Warning] uias:DESVersion attribute SHOULD be specified as version 202111 (Version:2021-NOV) with an optional extension. Found
    <sch:value-of select="@uias:DESVersion"/>.
    </sch:assert>
    </sch:rule>
</sch:pattern>
```

4.32 - ../Rules/UIAS_ID_00068.sch

Rule Description

[UIAS-ID-00068][Error] SCI control values (with at least one or more "-") must have its hierarchical parent value (without the last "-xxx") in the same list.

Code Description

Every SCI token in saml:AttributeValue that has at least one "-" must have its hierarchical parent value which is the substring before the last "-" be in the token list.

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="UIAS-ID-00068">
  <sch:rule id="UIAS-ID-00068-R1"
    context="saml:Attribute[@Name='fineAccessControls']">
    <sch:let name="tokenList"
      value="tokenize(normalize-space(string(../saml:AttributeValue)), ' ')" />
    <sch:assert test="every $evaluatedToken in $tokenList satisfies if (not(matches($evaluatedToken, 'SAR-')) and not(matches($evaluatedToken, 'NATO-')) and
matches($evaluatedToken, '-')) then util:existInTokenSet(util:substring-before-last($evaluatedToken, '-'), $tokenList) else true()"
      flag="error"
      role="error">[UIAS-ID-00068][Error] SCI control values (with at least one or more "-") must have its hierarchical parent value (without the last "-xxx") in
the same list. Some token in
    <sch:value-of select="$tokenList"/>is missing its hierarchical parent value (without the last "-xxx").
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

4.33 - .//Rules/UIAS_ID_00069.sch

Rule Description

[UIAS-ID-00069][Error] If the element with @Name-Role uses ENT namespace, the scope value must exist in the list of allowed ENT scope values.

Code Description

If element with @Name-Role uses ENT namespace, invoke abstract rule ValueExistsInList to check if the value exists in the list of allowed ENT scope values.

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="UIAS-ID-00069" is-a="ValidateTokenValuesExistInNamespaceList">
    <sch:param name="context" value="saml:Attribute[@Name='role']"/>
    <sch:param name="namespaceValue" value="saml:AttributeValue[1]"/>
    <sch:param name="namespaceUsed" value="'ENT'"/>
    <sch:param name="tokenList" value="$roleEnterpriseRoleList"/>
    <sch:param name="errMsg"
        value="'[UIAS-ID-00069][Error] If the element with @Name-Role uses ENT namespace, the scope value must exist in the list of allowed ENT scope values.'"/>
</sch:pattern>
```

4.34 - ../Rules/UIAS_ID_00070.sch

Rule Description

[UIAS-ID-00070][Error] If the element with @Name-Role uses Nebula namespace, the scope value must exist in the list of allowed Nebula scope values.

Code Description

If element with @Name-Role uses Nebula namespace, invoke abstract rule ValueExistsInList to check if the value exists in the list of allowed Nebula scope values.

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="UIAS-ID-00070" is-a="ValidateTokenValuesExistInNamespaceList">
    <sch:param name="context" value="saml:Attribute[@Name='role']"/>
    <sch:param name="namespaceValue" value="saml:AttributeValue[1]"/>
    <sch:param name="namespaceUsed" value="'Nebula'"/>
    <sch:param name="tokenList" value="$roleNebulaNamedRoleList"/>
    <sch:param name="errMsg"
        value="'[UIAS-ID-00070][Error] If the element with @Name-Role uses Nebula namespace, the scope value must exist in the list of allowed Nebula scope values.'"/>
</sch:pattern>
```

4.35 - ./Rules/UIAS_ID_00071.sch

Rule Description

[UIAS-ID-00071][Error] The saml:Attribute[@Name='topic']/saml:AttributeValue element MUST have "ANY". Human Readable: IF the 'Topic' attribute is present, the 'ANY' element MUST be provided.

Code Description

The saml:Attribute[@Name='topic']/saml:AttributeValue element MUST have at least 2 values. Human Readable: IF the 'Topic' attribute is present, there MUST be at least 2 values provided.

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="UIAS-ID-00071">
  <sch:rule id="UIAS-ID-00071-R1"
    context="saml:Attribute[@Name = 'topic']/saml:AttributeValue">
    <sch:let name="valueText" value="normalize-space(.)"/>
    <sch:assert test="contains($valueText, 'ANY')" flag="error" role="error">[UIAS-ID-00071][Error] IF the 'Topic' attribute is present, the 'ANY' element MUST be
provided.</sch:assert>
    <sch:assert test="count(tokenize($valueText, ' ')) > 1"
      flag="error"
      role="error">[UIAS-ID-00071][Error] IF the 'Topic' attribute is present, there MUST be at least 2 values provided.</sch:assert>
  </sch:rule>
</sch:pattern>
```

4.36 - ../Rules/UIAS_ID_00072.sch

Rule Description

[UIAS-ID-00072][Error] The saml:Attribute[@Name='region']/saml:AttributeValue element MUST have "ANAN". Human Readable: IF the 'Region' attribute is present, the 'ANAN' element MUST be provided.

Code Description

The saml:Attribute[@Name='region']/saml:AttributeValue element MUST have at least 2 values. Human Readable: IF the 'Region' attribute is present, there MUST be at least 2 values provided.

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="UIAS-ID-00072">
  <sch:rule id="UIAS-ID-00072-R1"
    context="saml:Attribute[@Name = 'region']/saml:AttributeValue">
    <sch:let name="valueText" value="normalize-space(.)"/>
    <sch:assert test="contains($valueText, 'ANAN')" flag="error" role="error">[UIAS-ID-00072][Error] IF the 'Region' attribute is present, the 'ANAN' element MUST be
provided.</sch:assert>
    <sch:assert test="count(tokenize($valueText, ' ')) > 1"
      flag="error"
      role="error">[UIAS-ID-00072][Error] IF the 'Region' attribute is present, there MUST be at least 2 values provided.</sch:assert>
  </sch:rule>
</sch:pattern>
```

4.37 - ../Rules/UIAS_ID_00073.sch

Rule Description

[UIAS-ID-00073][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'AUTHCAT' that is version '201804' (Version:2018-APR) 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="UIAS-ID-00073" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'201804'"/>
    <sch:param name="SpecToCheck" value="'AUTHCAT'"/>
    <sch:param name="pathToDocument" value="'../..../CVE/AUTHCAT/CVEnumAuthCatType.xml'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00073'"/>
</sch:pattern>
```

4.38 - ../Rules/UIAS_ID_00074.sch

Rule Description

[UIAS-ID-00074][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'FAC' that is version '201909' (Version:2019-SEP) 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="UIAS-ID-00074" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'201909'"/>
    <sch:param name="SpecToCheck" value="'FAC'"/>
    <sch:param name="pathToDocument"
        value="'../../CVE/FAC/CVEnumFineAccessControlType.xml'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00074'"/>
</sch:pattern>
```


4.39 - ../Rules/UIAS_ID_00075.sch

Rule Description

[UIAS-ID-00075][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'IC-GENC' that is version '201909' (Version:2019-SEP) 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="UIAS-ID-00075" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'201909'"/>
    <sch:param name="SpecToCheck" value="'IC-GENC'"/>
    <sch:param name="pathToDocument"
        value="'../../../../CVE/IC-GENC/CVEnumGENCCountryCode.xml'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00075'"/>
</sch:pattern>
```

4.40 - ../Rules/UIAS_ID_00076.sch

Rule Description

[UIAS-ID-00076][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'MN' that is version '201705.201903' (Version:2017-MAY - REVISION 2019-MAR) or later. NOTE: This is not an error of the instance document but of the validation environment itself.

Code Description

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

Schematron Code

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

<sch:pattern id="UIAS-ID-00076" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'201705.201903'"/>
    <sch:param name="SpecToCheck" value="'MN'"/>
    <sch:param name="pathToDocument" value="'../CVE/MN/CVEnumMNRRegion.xml'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00076'"/>
</sch:pattern>
```

4.41 - ../Rules/UIAS_ID_00077.sch

Rule Description

[UIAS-ID-00077][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'ROLE' 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="UIAS-ID-00077" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'202111'"/>
    <sch:param name="SpecToCheck" value="'ROLE'"/>
    <sch:param name="pathToDocument"
        value="'../../CVE/ROLE/CVEnumROLEC2SFunction.xml'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00077'"/>
</sch:pattern>
```

4.42 - ../Rules/UIAS_ID_00078.sch

Rule Description

[UIAS-ID-00078][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'USAgency' that is version '201703.201802' (Version:2017-MAR REVISION 2018-FEB) or later. NOTE: This is not an error of the instance document but of the validation environment itself.

Code Description

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

Schematron Code

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

<sch:pattern id="UIAS-ID-00078" is-a="ValidateValidationEnvCVE">
    <sch:param name="MinVersion" value="'201703.201802'"/>
    <sch:param name="SpecToCheck" value="'USAgency'"/>
    <sch:param name="pathToDocument"
        value="'../../CVE/USAgency/CVEnumUSAgencyAcronym.xml'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00078'"/>
</sch:pattern>
```

4.43 - ../Rules/UIAS_ID_00079.sch

Rule Description

[UIAS-ID-00079][Error] Regardless of the version indicated on the instance document, the validation infrastructure MUST use a version of 'VIRT' that is version '202010' (Version:2020-OCT) 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="UIAS-ID-00079" is-a="ValidateValidationEnvSchema">
    <sch:param name="MinVersion" value="'202010'"/>
    <sch:param name="SpecToCheck" value="'VIRT'"/>
    <sch:param name="pathToDocument" value="'../..Schema/VIRT/VIRT.xsd'"/>
    <sch:param name="RuleID" value="'UIAS-ID-00079'"/>
</sch:pattern>
```

4.44 - ./Rules/UIAS_ID_00080.sch

Rule Description

[UIAS-ID-00080][Error] All SAR tokens in fineAccessControls MUST conform to the regex ^SAR-[A-Z]{3,}:(C|S|TS){0,1}[A-Za-z0-9._-]{1,}\$. Human Readable: All SAR tokens in fineAccessControls must conform to a regular expression for: SAR-SourceAuthority:Classification:SAPmarking or SAR-SourceAuthority:SAPmarking.

Code Description

For all SAR tokens within fineAccessControls, this rule ensures that each token follows the regex ^SAR-[A-Z]{3,}:(C|S|TS){0,1}[A-Za-z0-9._-]{1,}\$. The rule looks for UIAS saml:Attribute elements that have attribute Name='fineAccessControls' and that contain SAR fineAccessControls identified by 'SAR-'. Next,, a variable \$nonmatchingTokens is created that takes all the SAR tokens in the saml

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="UIAS-ID-00080">
    <sch:rule id="UIAS-ID-00080-R1"
        context="saml:Attribute[@Name='fineAccessControls'][contains(./saml:AttributeValue,'SAR-')]">
        <sch:let name="nonmatchingTokens"
            value="for $token in tokenize(normalize-space(string(./saml:AttributeValue)), ' ') return if (starts-with($token,'SAR-') and not(matches($token,'^SAR-[A-Z]{3,}:(C|S|TS){0,1}[A-Za-z0-9._-]{1,}$')) then $token else null"/>
        <sch:assert test="count($nonmatchingTokens) = 0" flag="error" role="error">[UIAS-ID-00080][Error] All SAR tokens in the fineAccessControls attribute MUST conform to the
regex ^SAR-[A-Z]{3,}:(C|S|TS){0,1}[A-Za-z0-9._-]{1,}$ . Human Readable: All SAR tokens in fineAccessControls must conform to a regular expression for: SAR-
SourceAuthority:Classification:SAPmarking or SAR-SourceAuthority:SAPmarking.</sch:assert>
    </sch:rule>
</sch:pattern>
```

4.45 - ./Rules/UIAS_ID_00081.sch

Rule Description

[UIAS-ID-00081][Error] The SAR tokens in fineAccessControls must start with a substring after SAR- and before : that exists in the SAR Source Authorities CVE. Human Readable: The SAR tokens in fineAccessControls must contain a substring between the first and second colons in the pattern SAR:SourceAuthority:SARValue.

Code Description

For any SAR token within fineAccessControls, invoke ValidateEmbeddedValueExistenceinList to check that the token's substring after SAR- and before : exists in the SAR Source Authorities CVE.

Schematron Code

```
<?ICEA pattern?>
<?schematron-phases phaseids="VALUECHECK"?>
<!-- Original rule id: NTK-ID-00027 -->
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->

<sch:pattern id="UIAS-ID-00081" is-a="ValidateEmbeddedValueExistenceinList">
  <sch:param name="context"
    value="saml:Attribute[@Name='fineAccessControls'][contains(./saml:AttributeValue,'SAR-')]" />
  <sch:param name="searchTermList" value="./saml:AttributeValue" />
  <sch:param name="substringAfter" value="'SAR-'" />
  <sch:param name="substringBefore" value="':'" />
  <sch:param name="prefix" value="'SAR-'" />
  <sch:param name="list" value="$SARSourceAuthorityList" />
  <sch:param name="errMsg"
    value="'[UIAS-ID-00081][Error] The SAR tokens in fineAccessControls must start with a substring after SAR- and before : that exists in the SAR Source
Authorities CVE. Example from SAR tokens in fineAccessControls is SAR-DOD:TS:DEMOSAP1. The string DOD must be found in the SAR Source Authorities CVE. Human Readable: The SAR tokens in
fineAccessControls must contain a substring between after the SAR- and before the first : in the pattern SAR-SourceAuthority:Classification:SAPmarking or SAR-SourceAuthority-SARValue; this
substring must be found in the SAR Source Authorities CVE. Example SAR-DOD:TS:DEMOSAP1. The string DOD must be found in the SAR Source Authorities CVE. '" />
</sch:pattern>
```

Chapter 5 - Abstract Patterns

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

5.1 - `./Lib/ValidateEmbeddedValueExistenceinList.sch`

Code Description

This abstract pattern checks to see if an embedded string within an attribute of an element exists in a list or matches the pattern defined by the list. The calling rule must pass the context, search term list, attribute value to check, the substring to start after, the substring to stop before, a prefix, and an error message. The logic excludes tokens that do not start with \$prefix. Example from SAR tokens in fineAccessControls is SAR-DOD:TS:DEMOSAP1. The string DOD must be found in the SAR Source Authorities CVE.

Schematron Code

```
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<!--
  This abstract pattern checks to see if embedded strings in tokens with delimiting characters in an attribute of an element exists in a list or
  matches the pattern defined by the list.

  $context      := the context in which the searchValue exists
  $searchTermList := the set of values which you want to verify is in the list
  $list          := the list in which to search for the searchValue
  $substringAfter := a string that identifies the characters before the start of the value of interest
  $substringBefore:= a string that identifies the characters after the start of the value of interest
  $prefix        := a string that is used to filter the $searchTermList by selecting only those tokens that start with $prefix.
                  The rule does not apply to tokens that do NOT start with $prefix.
  $errMsg        := the error message text to display when the assertion fails

  Example usage:
  <sch:pattern is-a="ValidateValueExistenceInList" id="UIAS-ID-00081" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
    <sch:param name="context" value="saml:Attribute[@Name='fineAccessControls'][contains(/saml:AttributeValue,'SAR-'))]"/>
    <sch:param name="searchTermList" value="/saml:AttributeValue"/>
    <sch:param name="list" value="$SARSourceAuthorityList"/>
    <sch:param name="substringAfter" value="'SAR-'"
    <sch:param name="substringBefore" value="':'"
    <sch:param name="prefix" value="'SAR-'"
    <sch:param name="errMsg" value="'
      [ISM-ID-00530][Error] The SAR tokens in fineAccessControls must start with a substring before ':' and after 'SAR-' that exists
      in the SAR Source Authorities CVE. Example from SAR tokens in fineAccessControls is SAR-DOD:TS:DEMOSAP1. The string DOD must be
      found in the SAR Source Authorities CVE.
    '"/>
  </sch:pattern>

-->

<sch:pattern abstract="true" id="ValidateEmbeddedValueExistenceinList">
  <sch:rule id="ValidateEmbeddedValueExistenceInList-R1" context="$context">
    <sch:assert test="every $searchTerm in tokenize(normalize-space(string($searchTermList)), ' ') satisfies not(starts-with($searchTerm,$prefix)) or ((substring-
before((substring-after($searchTerm,$substringAfter)), $substringBefore)) = $list or (some $Term in $list satisfies (matches(substring-before(substring-after(normalize-space($searchTerm),
$substringAfter), $substringBefore), concat('^', $Term , '$')))))"
      flag="error"
      role="error">
      <sch:value-of select="$errMsg"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

5.2 - ./Lib/ValidateTokenValuesExistInNamespaceList.sch

Code Description

This abstract pattern verifies that a value at a given context matches a value in a list using regular expression matching. The calling rule must pass the context, search term list, attribute value to check, and an error message.

Schematron Code

```
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
      -->
<!--
      This abstract pattern verifies that a term matches a value in a list using regular expression matching.

      $context      := the context in which the searchValue exists
      $namespaceValue := the namespace value
      $namespaceUsed  := the namespace used
      $tokenList      := the list in which to search for the searchValue
      $errMsg         := the error message text to display when the assertion fails

      </sch:pattern>
-->

<sch:pattern abstract="true" id="ValidateTokenValuesExistInNamespaceList">
  <sch:rule id="ValidateTokenValuesExistInNamespaceList-R1" context="$context">
    <sch:assert test="every $token in tokenize(normalize-space(string($namespaceValue)), ' ') satisfies not(tokenize(string($token), '-') [1]=$namespaceUsed) or (every
$namespaceToken in tokenize(string($token), '-') [3] satisfies $namespaceToken = $tokenList or (some $item in $tokenList satisfies (matches(normalize-space($namespaceToken), concat('^',
$item, '$')))))"
      flag="error"
      role="error">
      <sch:value-of select="$errMsg"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

5.3 - ./Lib/ValidateTokenValuesExistenceInList.sch

Code Description

This abstract pattern checks to see if an attribute of an element exists in a list or matches the pattern defined by the list. The calling rule must pass the context, search term list, attribute value to check, and an error message.

Schematron Code

```
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<!--
      This abstract pattern checks to see if an attribute of an element exists in a list or matches the pattern defined by the list.

      $context      := the context in which the searchValue exists
      $searchTermList := the set of values which you want to verify is in the list
      $list          := the list in which to search for the searchValue
      $errMsg        := the error message text to display when the assertion fails

      Example usage:
      <sch:pattern is-a="ValidateValueExistenceInList" id="IRM_ID_00027" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:param name="context" value="@ism:releasableTo"/>
        <sch:param name="searchTermList" value="."/>
        <sch:param name="list" value="$releasableToList"/>
        <sch:param name="errMsg" value="'
          [ISM-ID-00265][Error] Any @ism:releasableTo must
          be a value in CVEnumISMRelTo.xml.
        '"/>
      </sch:pattern>

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

<sch:pattern abstract="true" id="ValidateTokenValuesExistenceInList">
  <sch:rule id="ValidateTokenValuesExistenceInList-R1" context="$context">
    <sch:assert test="every $token in tokenize(normalize-space(string($searchTermList)), ' ') satisfies $token = $list or (some $item in $list satisfies (matches(normalize-
space($token), concat('^',$item,'$'))))"
      flag="error"
      role="error">
      <sch:value-of select="$errMsg"/>
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

5.4 - ./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>
```

5.5 - ./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>
```

5.6 - ./Lib/ValidateValueExistenceInList.sch

Code Description

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

Schematron Code

```
<!-- Notices - Distribution Notice:
      This document has been approved for Public Release and is available for use without restriction.
-->
<!--
      This abstract pattern checks to see if an attribute of an element exists in a list.

      $context      := the context in which the searchValue exists
      $searchTerm    := the value which you want to verify is in the list
      $list          := the list in which to search for the searchValue
      $errMsg        := the error message text to display when the assertion fails

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

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

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

Chapter 6 - Schematron Schema

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

6.1 - `./UIAS_XML.sch`

Code Description

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

Schematron Code

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

<sch:schema queryBinding="xslt2">
    <sch:ns uri="urn:us:gov:ic:uias" prefix="uias"/>
    <sch:ns uri="urn:us:gov:ic:ism" prefix="ism"/>
    <sch:ns uri="urn:us:gov:ic:cve" prefix="cve"/>
    <sch:ns uri="urn:us:gov:ic:usagency" prefix="usagency"/>
    <sch:ns uri="urn:us:gov:ic:virt" prefix="virt"/>
    <sch:ns uri="urn:us:gov:ic:mn" prefix="mn"/>
    <sch:ns uri="urn:oasis:names:tc:SAML:2.0:assertion" prefix="saml"/>
    <sch:ns uri="http://www.w3.org/2001/XMLSchema-instance" prefix="xsi"/>
    <sch:ns uri="urn:us:gov:ic:uias:xsl:util" prefix="util"/>
    <!-- (U) Abstract Patterns -->

    <sch:include href="Lib/ValidateValueExistenceInList.sch"/>
    <sch:include href="Lib/ValidateEmbeddedValueExistenceinList.sch"/>
    <sch:include href="Lib/ValidateTokenValuesExistenceInList.sch"/>
    <sch:include href="Lib/ValidateTokenValuesExistInNamespaceList.sch"/>
    <sch:include href="Lib/ValidateValidationEnvCVE.sch"/>
    <sch:include href="Lib/ValidateValidationEnvSchema.sch"/>
    <!-- (U) Functions -->

    <xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        name="util:substring-before-last">
        <xsl:param name="input" as="xs:string?" />
        <xsl:param name="delimiter" as="xs:string" />
        <xsl:value-of select="codepoints-to-string(reverse(string-to-codepoints(substring-after(codepoints-to-string(reverse(string-to-codepoints($input))), $delimiter))))" />
    </xsl:function>
    <xsl:function xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        name="util:existInTokenSet"
        as="xs:boolean">
        <xsl:param name="stringTokenValue" />
        <xsl:param name="tokenList" as="xs:string*" />
        <xsl:value-of select="tokenize($stringTokenValue, ' ') = $tokenList" />
    </xsl:function>
    <!-- (U) Resources -->

    <sch:let name="certificateAuthorityList"
        value="document(' ../../CVE/UIAS/CVEnumUIASCertificateAuthority.xml ' )//cve:CVE/cve:Enumeration/cve:Term/cve:Value" />
    <sch:let name="clearanceList"
        value="document(' ../../CVE/UIAS/CVEnumUIASClearance.xml ' )//cve:CVE/cve:Enumeration/cve:Term/cve:Value" />
    <sch:let name="entityTypeList"
```

```

        value="document('.../CVE/UIAS/CVEnumUIASEntityType.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="personEntityList"
        value="document('.../CVE/UIAS/CVEnumUIASPersonEntityType.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="nonPersonEntityList"
        value="document('.../CVE/UIAS/CVEnumUIASNonPersonEntityType.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="lifeCycleStatusList"
        value="document('.../CVE/UIAS/CVEnumUIASLifeCycleStatus.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="handlingControlsList"
        value="document('.../CVE/UIAS/CVEnumUIASHandlingControls.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="usAgencyList"
        value="document('.../CVE/USAgency/CVEnumUSAgencyAcronym.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="facList"
        value="document('.../CVE/FAC/CVEnumFineAccessControlType.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="SARSourceAuthorityList"
        value="document('.../CVE/FAC/CVEnumFACSARAAuthorities.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="authcatList"
        value="document('.../CVE/AUTHCAT/CVEnumAuthCatType.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="regionList"
        value="document('.../CVE/MN/CVEnumMNRegion.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="topicList"
        value="document('.../CVE/MN/CVEnumMNIssue.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="respEntList"
        value="document('.../CVE/ISM/CAT/CVEnumISM/CATResponsibleEntity.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="networkList"
        value="document('.../CVE/VIRT/CVEnumVIRTNetworkName.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="dissemList"
        value="document('.../CVE/ISM/CVEnumISMDissem.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="auditRoutingOrgList"
        value="document('.../CVE/USAgency/CVEnumAuditRoutingOrg.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="schemaTypeList"
        value="document('.../CVE/UIAS/CVEnumUIASSchemaType.xml')//cve:CVE/cve:Enumeration/cve:Term"/>
<sch:let name="roleC2SScopeList"
        value="document('.../CVE/ROLE/CVEnumROLEC2SScope.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="rolePAASScopeList"
        value="document('.../CVE/ROLE/CVEnumROLEPAASScope.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="roleNebulaNamedRoleList"
        value="document('.../CVE/ROLE/CVEnumROLENebulaNamedRole.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="roleEnterpriseRoleList"
        value="document('.../CVE/ROLE/CVEnumROLEEnterpriseRole.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="roleNameSpaceList"
        value="document('.../CVE/ROLE/CVEnumROLENamespace.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="roleC2SFunctionList"
        value="document('.../CVE/ROLE/CVEnumROLEC2SFunction.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<sch:let name="rolePAASFunctionList"
        value="document('.../CVE/ROLE/CVEnumROLEPAASFunction.xml')//cve:CVE/cve:Enumeration/cve:Term/cve:Value"/>
<!-- ***** -->
<!-- (U) UIAS Phases -->
<!-- ***** -->
<!-- ***** -->
<!-- (U) UIAS ID Rules -->
<!-- ***** -->
<!--(U) -->

<sch:include href="./Rules/UIAS_ID_00001.sch"/>
```

```
<sch:include href="./Rules/UIAS_ID_00004.sch"/>
<sch:include href="./Rules/UIAS_ID_00005.sch"/>
<sch:include href="./Rules/UIAS_ID_00006.sch"/>
<sch:include href="./Rules/UIAS_ID_00007.sch"/>
<sch:include href="./Rules/UIAS_ID_00008.sch"/>
<sch:include href="./Rules/UIAS_ID_00009.sch"/>
<sch:include href="./Rules/UIAS_ID_00011.sch"/>
<sch:include href="./Rules/UIAS_ID_00012.sch"/>
<sch:include href="./Rules/UIAS_ID_00014.sch"/>
<sch:include href="./Rules/UIAS_ID_00016.sch"/>
<sch:include href="./Rules/UIAS_ID_00019.sch"/>
<sch:include href="./Rules/UIAS_ID_00021.sch"/>
<sch:include href="./Rules/UIAS_ID_00022.sch"/>
<sch:include href="./Rules/UIAS_ID_00023.sch"/>
<sch:include href="./Rules/UIAS_ID_00024.sch"/>
<sch:include href="./Rules/UIAS_ID_00025.sch"/>
<sch:include href="./Rules/UIAS_ID_00026.sch"/>
<sch:include href="./Rules/UIAS_ID_00028.sch"/>
<sch:include href="./Rules/UIAS_ID_00030.sch"/>
<sch:include href="./Rules/UIAS_ID_00036.sch"/>
<sch:include href="./Rules/UIAS_ID_00047.sch"/>
<sch:include href="./Rules/UIAS_ID_00050.sch"/>
<sch:include href="./Rules/UIAS_ID_00051.sch"/>
<sch:include href="./Rules/UIAS_ID_00052.sch"/>
<sch:include href="./Rules/UIAS_ID_00053.sch"/>
<sch:include href="./Rules/UIAS_ID_00056.sch"/>
<sch:include href="./Rules/UIAS_ID_00057.sch"/>
<sch:include href="./Rules/UIAS_ID_00065.sch"/>
<sch:include href="./Rules/UIAS_ID_00066.sch"/>
<sch:include href="./Rules/UIAS_ID_00067.sch"/>
<sch:include href="./Rules/UIAS_ID_00068.sch"/>
<sch:include href="./Rules/UIAS_ID_00069.sch"/>
<sch:include href="./Rules/UIAS_ID_00070.sch"/>
<sch:include href="./Rules/UIAS_ID_00071.sch"/>
<sch:include href="./Rules/UIAS_ID_00072.sch"/>
<sch:include href="./Rules/UIAS_ID_00073.sch"/>
<sch:include href="./Rules/UIAS_ID_00074.sch"/>
<sch:include href="./Rules/UIAS_ID_00075.sch"/>
<sch:include href="./Rules/UIAS_ID_00076.sch"/>
<sch:include href="./Rules/UIAS_ID_00077.sch"/>
<sch:include href="./Rules/UIAS_ID_00078.sch"/>
<sch:include href="./Rules/UIAS_ID_00079.sch"/>
<sch:include href="./Rules/UIAS_ID_00080.sch"/>
<sch:include href="./Rules/UIAS_ID_00081.sch"/>
<!-- *****-->
<!-- (U) UIAS Phases -->
<!-- *****-->
<!--(U) Phase: Non-infrastructure-->

<sch:phase id="NON_INFRASTRUCTURE">
    <sch:active pattern="UIAS-ID-00081"/>
</sch:phase>
<!--(U) Phase: VALUECHECK-->
```

```
<sch:phase id="VALUECHECK">
    <sch:active pattern="UIAS-ID-00081"/>
</sch:phase>
</sch:schema>
<!--UNCLASSIFIED-->
```

Chapter 7 - Removed Rules

All of the numbered Rules for UIAS 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.

7.1 - `./Rules/deleted/UIAS_ID_00002.sch`

Rule Description

[UIAS-ID-00002][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.2 - `./Rules/deleted/UIAS_ID_00003.sch`

Rule Description

[UIAS-ID-00003][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.3 - `./Rules/deleted/UIAS_ID_00010.sch`

Rule Description

[UIAS-ID-00010][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.4 - `./Rules/deleted/UIAS_ID_00013.sch`

Rule Description

[UIAS-ID-00013][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.5 - `./Rules/deleted/UIAS_ID_00015.sch`

Rule Description

[UIAS-ID-00015][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.6 - `./Rules/deleted/UIAS_ID_00017.sch`

Rule Description

[UIAS-ID-00017][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.7 - `./Rules/deleted/UIAS_ID_00018.sch`

Rule Description

[UIAS-ID-00018][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.8 - **./Rules/deleted/UIAS_ID_00020.sch**

Rule Description

[UIAS-ID-00020][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.9 - **./Rules/deleted/UIAS_ID_00027.sch**

Rule Description

[UIAS-ID-00027][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.10 - **./Rules/deleted/UIAS_ID_00029.sch**

Rule Description

[UIAS-ID-00029][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.11 - **./Rules/deleted/UIAS_ID_00031.sch**

Rule Description

[UIAS-ID-00031][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.12 - **./Rules/deleted/UIAS_ID_00032.sch**

Rule Description

[UIAS-ID-00032][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.13 - **./Rules/deleted/UIAS_ID_00033.sch**

Rule Description

[UIAS-ID-00033][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.14 - **./Rules/deleted/UIAS_ID_00034.sch**

Rule Description

[UIAS-ID-00034][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.15 - **./Rules/deleted/UIAS_ID_00035.sch**

Rule Description

[UIAS-ID-00035][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.16 - **./Rules/deleted/UIAS_ID_00037.sch**

Rule Description

[UIAS-ID-00037][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.17 - **./Rules/deleted/UIAS_ID_00038.sch**

Rule Description

[UIAS-ID-00038][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.18 - **./Rules/deleted/UIAS_ID_00039.sch**

Rule Description

[UIAS-ID-00039][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.19 - **./Rules/deleted/UIAS_ID_00040.sch**

Rule Description

[UIAS-ID-00040][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.20 - **./Rules/deleted/UIAS_ID_00041.sch**

Rule Description

[UIAS-ID-00041][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.21 - **./Rules/deleted/UIAS_ID_00042.sch**

Rule Description

[UIAS-ID-00042][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.22 - **./Rules/deleted/UIAS_ID_00043.sch**

Rule Description

[UIAS-ID-00043][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.23 - **./Rules/deleted/UIAS_ID_00044.sch**

Rule Description

[UIAS-ID-00044][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.24 - **./Rules/deleted/UIAS_ID_00045.sch**

Rule Description

[UIAS-ID-00045][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.25 - **./Rules/deleted/UIAS_ID_00046.sch**

Rule Description

[UIAS-ID-00046][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.26 - **./Rules/deleted/UIAS_ID_00048.sch**

Rule Description

[UIAS-ID-00048][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.27 - **./Rules/deleted/UIAS_ID_00049.sch**

Rule Description

[UIAS-ID-00049][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.28 - **./Rules/deleted/UIAS_ID_00054.sch**

Rule Description

[UIAS-ID-00054][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.29 - **./Rules/deleted/UIAS_ID_00055.sch**

Rule Description

[UIAS-ID-00055][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.30 - **./Rules/deleted/UIAS_ID_00058.sch**

Rule Description

[UIAS-ID-00058][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.31 - **./Rules/deleted/UIAS_ID_00059.sch**

Rule Description

[UIAS-ID-00059][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.32 - **./Rules/deleted/UIAS_ID_00060.sch**

Rule Description

[UIAS-ID-00060][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.33 - **./Rules/deleted/UIAS_ID_00061.sch**

Rule Description

[UIAS-ID-00061][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.34 - **./Rules/deleted/UIAS_ID_00062.sch**

Rule Description

[UIAS-ID-00062][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.35 - **./Rules/deleted/UIAS_ID_00063.sch**

Rule Description

[UIAS-ID-00063][Error] Introduced and removed in V2016-SEP. Never made it to signed package.

7.36 - `./Rules/deleted/UIAS_ID_00064.sch`

Rule Description

[UIAS-ID-00064][Error] Introduced and removed in V2016-SEP. Never made it to signed package.