

Conformance Targets Attribute Specification (CTAS) Version 3.0

Project Specification 01

22 February 2023

This stage:

<https://docs.oasis-open.org/niemopen/ctas/v3.0/ps01/ctas-v3.0-ps01.html> (Authoritative)

<https://docs.oasis-open.org/niemopen/ctas/v3.0/ps01/ctas-v3.0-ps01.pdf>

Previous stage:

<https://docs.oasis-open.org/niemopen/ctas/v3.0/psd01/ctas-v3.0-psd01.html> (Authoritative)

<https://docs.oasis-open.org/niemopen/ctas/v3.0/psd01/ctas-v3.0-psd01.pdf>

Latest stage:

<https://docs.oasis-open.org/niemopen/ctas/v3.0/ctas-v3.0.html> (Authoritative)

<https://docs.oasis-open.org/niemopen/ctas/v3.0/ctas-v3.0.pdf>

Open Project:

[OASIS NIEMOpen OP](#)

Project Chair:

Katherine Escobar (katherine.b.escobar.civ@mail.mil), [Joint Staff J6](#)

NIEM Technical Architecture Committee (NTAC) Chairs:

Scott Renner (sar@mitre.org), [MITRE](#)

Jim Cabral (jim.cabral@infotrack.com), [InfoTrack](#)

Editor:

Tom Carlson (Thomas.Carlson@gtri.gatech.edu), [GTRI](#)

Additional artifacts:

This document is one component of a Work Product that also includes:

- XML schema: <https://docs.oasis-open.org/niemopen/ctas/v3.0/ps01/schemas/conformanceTargets.xsd>

Related work:

This specification is related to:

- *NIEM Code Lists Specification*. <https://reference.niem.gov/niem/specification/code-lists/4.0.1/niem-code-lists-spec.html>
- *NIEM Conformance Specification*. <https://reference.niem.gov/niem/specification/conformance/5.0/niem->

[conformance-spec-5.0.html](#)

- *NIEM Information Exchange Package Documentation Specification.*
<https://reference.niem.gov/niem/specification/model-package-description/5.0/niem-iepd-spec-5.0.html>

Declared XML namespace:

- <http://release.niem.gov/niem/conformanceTargets/3.0/>

Abstract:

This document specifies an XML attribute that may occur within XML documents to establish a claim that the document conforms to a set of conformance targets. This specification does not define what the conformance targets are; the conformance targets are established by other specifications. Each conformance target is identified by an internationalized resource identifier, which is defined by its relevant specification.

Status:

This document was last revised or approved by the Project Governing Board of the OASIS NIEMOpen OP on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Open Project (OP) are listed at <https://www.niemopen.org/>.

Comments on this work can be provided by opening issues in the project repository or by sending email to the project's public comment list: niemopen-comment@lists.oasis-open-projects.org. List information is available at <https://lists.oasis-open-projects.org/g/niemopen-comment/topics>.

Note that any machine-readable content ([Computer Language Definitions](#)) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Key words:

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#)] and [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

Citation format:

When referencing this specification the following citation format should be used:

[CTAS-v3.0]

Conformance Targets Attribute Specification (CTAS) Version 3.0. Edited by Tom Carlson. 22 February 2023. OASIS Project Specification 01. <https://docs.oasis-open.org/niemopen/ctas/v3.0/ps01/ctas-v3.0-ps01.html>. Latest stage: <https://docs.oasis-open.org/niemopen/ctas/v3.0/ctas-v3.0.html>.

Notices

Copyright © OASIS Open 2023. All Rights Reserved.

Distributed under the terms of the OASIS [IPR Policy](#).

For complete copyright information please see the Notices section in [Appendix H](#).

Table of Contents

- [1. Introduction](#)
 - [1.1. Background: NIEM 2 Conformant Indicator](#)
 - [1.2. Goals of This Specification](#)
 - [1.3. Description of the Conformance Targets Attribute](#)
- [2. Document Conventions](#)
 - [2.1. RFC 2119 Terminology](#)
 - [2.2. Use of Namespaces](#)
 - [2.3. Clark Notation](#)
 - [2.4. Normative content](#)
- [3. Conformance](#)
 - [3.1. Conformance Target Defined](#)
 - [3.2. Conformance to this Specification](#)
 - [3.3. Syntactic Conformance](#)
- [4. Semantics and Use](#)
 - [4.1. Suggested Validator Behavior](#)
 - [4.2. Use of this Specification](#)
- [Appendix A. References](#)
- [Appendix B. XML Schema Document for Conformance Targets Attribute](#)
- [Appendix C. Sample XSLT Version 1.0 Implementation](#)
- [Appendix D. Sample XSLT Version 2.0 Implementation](#)
- [Appendix E. Index of definitions](#)
- [Appendix F. Index of rules](#)
- [Appendix G. Acknowledgments](#)
- [Appendix H. Notices](#)

Table of Figures

- [Figure 1-1: Example of NIEM 2 ConformantIndicator](#)
- [Figure 1-2: Example of conformance target attribute in an XML Schema document](#)
- [Figure 2-1: Clark notation for `xs:string`](#)
- [Figure 4-1: XPath 2 expression yielding effective conformance targets attribute](#)
- [Figure B-1: XML Schema Document for Conformance Targets Attribute](#)
- [Figure C-1: Sample XSLT Version 1.0 Implementation](#)
- [Figure D-1: Sample XSLT Version 2.0 Implementation](#)

1. Introduction

The Conformance Targets Attribute Specification defines an attribute that, when it appears in an XML document, claims the document conforms to one or more conformance targets. This pattern and specification was developed to overcome shortcomings in the NIEM 2 `ConformantIndicator` element, and to provide needed capabilities in future specifications.

1.1. Background: NIEM 2 Conformant Indicator

Under NIEM 2, a NIEM-conformant XML schema document must indicate that it is a NIEM-conformant schema. This is done through the occurrence of an element within `xs:appinfo` in the schema:

Figure 1-1: Example of NIEM 2 ConformantIndicator

```
<xs:schema [...]>
  <xs:annotation>
    <xs:documentation>[...]</xs:documentation>
    <xs:appinfo>
      <i:ConformantIndicator>true</i:ConformantIndicator>
    </xs:appinfo>
  </xs:annotation>
  [...]
</xs:schema>
```

The NIEM 2 conformance indicator is limited:

1. The NIEM 2 conformance indicator does not work in documents other than XML Schema documents. It is defined to only apply within a schema document, when the conformant indicator is an `appinfo` on the document element (root element) of the schema document.
2. The NIEM 2 conformance indicator only identifies conformance to a single standard, NIEM 2, as indicated by the namespace in which the `appinfo:ConformantIndicator` element is defined.
3. The NIEM 2 conformance indicator does not explicitly identify a conformance target. It indicates that a schema document claims to be conformant, but does not indicate to which conformance target it conforms.

1.2. Goals of This Specification

The construct specified by this specification has the following goals:

1. It supports specifying conformance targets within any XML document, not just XML Schema documents. Any future XML document may be designed to accommodate it.
2. It places a minimum of syntactic requirements on an XML document. It does not require a particular document element. It does not require an attribute directly on the document element.
3. It does not introduce unnecessary content, including wildcards in a content model.
4. It does not introduce dependencies on unnecessary architectural components or schemas.
5. It can express a claim to conform to a conformance target defined by a subsequent specification, without modification to this specification.
6. Subsequent specifications may explicitly leverage it with minimal effort.
7. It does not require any sort of registry or centrally-controlled mechanism.
8. It may be used by validation software to identify the correct validation criteria for an XML document.
9. It can express claims to multiple conformance targets simultaneously.
10. It supports nested documents, each with its own conformance claim, without conflict.
11. It may be validated with XML Schema, but does not require its use.
12. It may be used with XSLT 1.0 ([\[XSLT1\]](#)) and XSLT 2.0 ([\[XSLT2\]](#)) stylesheets.

1.3. Description of the Conformance Targets Attribute

This specification normatively defines a conformance target attribute. An XML document, for example a NIEM XML Schema document, will contain this attribute to indicate to which conformance targets it claims to conform. The attribute is a claim of conformance, and not a statement that should be trusted by a validating system. A validator would use this claim to identify to which conformance rules a document should be validated.

The attribute's value is a list of internationalized resource identifiers (IRIs). A later specification may define an IRI for its conformance target, and when an XML document has that IRI in its conformance target attribute, the document is claiming to conform to that conformance target.

Note that all examples within this document, unless explicitly labeled otherwise, are not intended to represent any actual standard, version, conformance target, or identifier. All examples are mock-ups, using familiar terms and subjects, but should not be interpreted to actually represent other specifications or activities.

As mocked-up example, a specification, *NIEM Business Rules Specification, Version 3.0* may define a conformance target called a “conformant schema document”. Note that the conformance target is particular to the specification, and the single version of that specification. In this case, an XML document that conforms to that conformance target would be an XML Schema document that follows the set of rules defined by that specification for that conformance target. Another specification may refer to that conformance target using text similar to the following:

... the document must be a *conformant schema document*, as defined by the *NIEM Business Rules Specification, Version 3.0* ...

The *NIEM Business Rules Specification, Version 3.0* would define an IRI that uniquely identifies that conformance target, for example:

`http://reference.niem.gov/niem/specification/business-rules/3.0/#SchemaDocument`

The IRI identifies an authority, a specification, a particular version of that specification, and a particular conformance target within that version of the specification.

A schema document could claim to conform to that conformance target by listing the IRI that is the identifier of the business rules “conformant schema document” conformance target in its first conformance targets attribute, as follows:

Figure 1-2: Example of conformance target attribute in an XML Schema document

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ct="http://release.niem.gov/niem/conformanceTargets/3.0/"
  ct:conformanceTargets="
    http://reference.niem.gov/niem/specification/naming-and-design-
rules/3.0/#ReferenceSchemaDocument
    http://reference.niem.gov/niem/specification/business-rules/3.0/#SchemaDocument"
  [...]>
<xs:annotation>
  <xs:documentation>[...]</xs:documentation>
</xs:annotation>
[...]
```

This example shows two conformance targets (again, this should not be read as reflecting actual standards, conformance targets, or identifiers):

- The first conformance target (`...#ReferenceSchemaDocument`) establishes a claim that the document conforms to the NIEM NDR, version 3.0's reference schema document conformance target.
- The second conformance target (`...#SchemaDocument`) establishes a claim that the document conforms to a NIEM business rules specification, version 3.0's schema document conformance target.

Standards Track Work Product

This conformance target claim is just that, a claim. It does not establish conformance, merely hints at it. The conformance claim may be overridden in many ways, such as a well-known root element, doctype, schema, etc. In the above example, the document element `xs:schema` indicates that the document should be treated as an XML Schema document. The conformance targets attribute indicates conformance targets beyond what is identified by the `xs:schema` element.

As another example, a ULEX 1.0 message has a well-known document element, `{ulex:message:pd:1.0}doPublish`, which indicates a ULEX conformance target and indicates use of ULEX's rules for validation. In addition, a specification that uses ULEX may choose to leverage this conformance targets attribute specification to specify additional conformance targets, by including within a message a `ct:conformanceTargets` attribute that indicates additional conformance targets.

In addition, an XML document may have an implied semantic that some specific sub-element (e.g., a ULEX `StructuredPayload` element) may contain an XML element that is meant to represent a separate XML document. In such a case, the XML fragment may be written out as a separate document, and the first conformance targets attribute within that document will establish a claim to conformance. Validation of such nested documents could be performed by a smart validator, without writing out sub-documents, but specification of that is outside the scope of this specification.

In any case, the first occurrence in an XML document of the attribute `ct:conformanceTargets` (in the namespace `http://release.niem.gov/niem/conformanceTargets/3.0/`) is termed the document's *effective conformance targets attribute* (see [\[Definition: effective conformance targets attribute\]](#)). The IRIs carried within that attribute are the document's *effective conformance target identifiers* (see [\[Definition: effective conformance target identifier\]](#)). Each of these identifiers is an IRI that indicates a conformance target to which the document may conform.

While this mechanism is available to specifications as a means of expressing a document's claim to conform, different specifications and document types may have different ways of specifying conformance. For example, the MPD catalog may use the attribute to express conformance of the catalog itself, while defining a separate mechanism to indicate the conformants of a set of documents, since a set is not itself an XML document.

This specification does not establish a requirement that any document use the conformance attribute to identify its conformance targets. The authors of subsequent specifications should make explicit whether the conformance attribute is a required mechanism for labeling conformance, or if it merely is used as an optional hint.

2. Document Conventions

This document uses formatting and syntactic conventions to clarify meaning and to avoid ambiguity.

2.1. RFC 2119 Terminology

Within normative content (rules and definitions), the key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in [\[RFC2119\]](#).

2.2. Use of Namespaces

The following namespaces are used consistently within this document:

- The namespace prefix `xs` is bound to the namespace URI reference `http://www.w3.org/2001/XMLSchema`.
- The namespace prefix `ct` is bound to the namespace URI reference `http://release.niem.gov/niem/conformanceTargets/3.0/`.

The use of namespace prefixes and prefix-qualified names within this specification does not constitute a requirement for any artifact to use any specific namespace prefix. Namespace prefixes are immaterial to conformance to this specification.

2.3. Clark Notation

This document uses Clark notation to represent qualified names in normative text. Clark notation is described by [\[ClarkNS\]](#), and provides the information in an XML qualified name (as defined by [\[XMLNamespaces\]](#)) without the need to define a namespace prefix and then reference that namespace prefix. A Clark notation value looks like the following:

Figure 2-1: Clark notation for `xs:string`

```
{http://www.w3.org/2001/XMLSchema}string
```

Each Clark notation value consists of a namespace URI surrounded by curly braces, concatenated with a local name. Such a namespace/local name pair may be expressed in XML using a namespace prefix, or a default namespace binding, as specified in the [\[XMLNamespaces\]](#).

2.4. Normative content

This document contains text that is normative, and text that is informative. This section describes the normative forms.

A definition defines a term that is used within rules.

[Definition: <term>]

A formal definition of a term used within the document.

A rule states a specific requirement on a conformance target. The conformance target is explicitly identified in the text of the rule. A rule uses terminology from [\[RFC2119\]](#) to establish requirements. Rules are normative.

[Rule <section>-<number>]

An enforceable rule.

3. Conformance

3.1. Conformance Target Defined

This document uses the term *conformance target* in a normative way:

[Definition: conformance target]

A *conformance target* is a class of artifact, such as an interface, protocol, document, platform, process or service, that is the subject of conformance clauses and normative statements. There may be several conformance targets defined within a specification, and these targets may be diverse so as to reflect different aspects of a specification. For example, a protocol message and a protocol engine may be different conformance targets.

The above definition of *conformance target* is adapted from [\[OASISConformance\]](#). This definition is overly broad for the purposes of this specification, as this specification is concerned only with conformance targets that are XML documents.

[Definition: conformance target identifier]

A *conformance target identifier* is an internationalized resource identifier that uniquely identifies a conformance target.

The term *internationalized resource identifier* (IRI) is defined by [\[RFC3987\]](#). The RFC also defines comparison of IRIs.

3.2. Conformance to this Specification

This specification defines a single conformance target, normatively referred to within this document as a *conformant document*.

[Definition: conformant document]

A *conformant document* is a document that uses the conformance targets attribute defined by this specification to express a claim that it conforms to specific conformance targets. A conformant document MUST satisfy all rules in this specification for conformant documents.

For an example of the use of this term in another specification, see [Section 4.2, Use of this Specification, below](#).

3.3. Syntactic Conformance

Rule 3-1. Conformant document is an XML document

[Rule 3-1]

A conformant document MUST be an XML document.

The term *XML document* is defined by [\[XML\]](#).

Rule 3-2. Attribute `ct:conformanceTargets` is valid**[Rule 3-2]**

Within a conformant document, every occurrence of the attribute `{http://release.niem.gov/niem/conformanceTargets/3.0/}conformanceTargets` **MUST** be locally schema-valid to the XML Schema component definition contained in [Figure B-1, XML Schema Document for Conformance Targets Attribute](#).

The schema-validity of an attribute is defined by [\[XMLSchema1\]](#). This specification does not require that the validity of a conformant artifact actually be assessed, only that the occurrences of the attribute satisfy the constraints embodied in the schema document.

This specification does not require any specific system or exchange specification to use this particular schema document. A system or exchange specification may use or incorporate a schema for this namespace that adds or removes annotations or modifies other characteristics of the schema, for example to make the schema document conformant to some specification.

Rule 3-3. Namespace defines only attribute `ct:conformanceTargets`**[Rule 3-3]**

A conformant document **MUST NOT** contain any element or attribute information item that has the namespace name `http://release.niem.gov/niem/conformanceTargets/3.0/`, other than attribute `{http://release.niem.gov/niem/conformanceTargets/3.0/}conformanceTargets`.

Rule 3-4. No reference to types in conformance targets namespace**[Rule 3-4]**

A conformant document **MUST NOT** contain an attribute `{http://www.w3.org/2001/XMLSchema-instance}type` with a value that has a namespace name of `http://release.niem.gov/niem/conformanceTargets/3.0/`.

[Rule 3-3. Namespace defines only attribute `ct:conformanceTargets`, above](#), and [Rule 3-4. No reference to types in conformance targets namespace, above](#), defend against abuse of the conformance targets attribute namespace, ensuring that any document that implies that there are additional schema components defined in the conformance targets attribute namespace is not conformant.

Rule 3-5. Attribute `ct:conformanceTargets` contains only absolute IRI references**[Rule 3-5]**

Within a conformant document, each item in the value of an attribute `{http://release.niem.gov/niem/conformanceTargets/3.0/}conformanceTargets` **MUST** be an absolute IRI reference.

The definition of an absolute IRI reference is defined by [\[RFC3987\]](#).

Rule 3-6. Document may contain any number of `ct:conformanceTargets` attributes

[Rule 3-6]

A conformant document MAY include any number of occurrences of the attribute
{<http://release.niem.gov/niem/conformanceTargets/3.0/>}conformanceTargets.

This specification does not restrict the number of occurrences of the conformance targets attribute. If a document has no occurrences of the conformance targets attribute, or if the effective conformance target attribute contains no items, then the document does not claim to conform to any conformance targets via the mechanisms of this specification. If there are multiple occurrences of the conformance targets attribute, only one is effective for the document.

4. Semantics and Use

This section defines the meaning of the constructs, and provides text for other specifications to leverage the Conformance Targets Attribute Specification.

[Definition: effective conformance targets attribute]

The *effective conformance targets attribute* of a conformant document is the first occurrence of the attribute `{http://release.niem.gov/niem/conformanceTargets/3.0/}conformanceTargets`, in document order.

The term *document order* is defined by [XPathDMJ](#). A suggested XPath 2 expression for the effective conformance targets attribute follows. It is not normative.

Figure 4-1: XPath 2 expression yielding effective conformance targets attribute

```
(//@*:conformanceTargets[
  namespace-uri() = 'http://release.niem.gov/niem/conformanceTargets/3.0/'
])[1]
```

An IRI reference that occurs within the effective conformance targets attribute establishes a claim to conform to some conformance target. Each conformance target attribute may make any number of claims. Each such claim may be testable by validation software or by human review to verify or falsify the claim that it conforms. Each such identifier is an *effective conformance target identifier*. Validation of a claim to conform to a conformance target is outside the scope of this specification.

[Definition: effective conformance target identifier]

An *effective conformance target identifier* of a conformant document is an internationalized resource identifier reference that occurs in the document's effective conformance targets attribute.

It is expected (but not required) that each effective conformance target identifier will be an IRI that is defined by some specification to represent a conformance target.

4.1. Suggested Validator Behavior

This section includes a non-normative description of the way the constructs defined by this specification are intended to work. This section is not normative.

A validator should use the effective conformance target identifiers of a conformant document as hints indicating possible intended conformance targets for the document.

A validator should not identify claimed conformance targets using occurrences of the conformance targets attribute other than the effective conformance targets attribute. Only the first occurrence of the conformance targets attribute, in document order, establishes a claim to conformance.

A validator should limit validation to known conformance target identifiers. It should ignore unknown identifiers.

A validator should not treat the order of conformance target identifiers within the effective conformance targets attribute as significant. The contents of a conformance targets attribute should be treated as a set, rather than an ordered list.

4.2. Use of this Specification

Standards Track Work Product

The following should be true for a specification that leverages this specification:

1. The specification defines explicit conformance targets.
2. Normative statements made by the specification are prescriptive requirements on those conformance targets.
3. Each conformance target that is an XML document, and for which the Conformance Targets Attribute Specification will be leveraged, is assigned a conformance target identifier by the specification. Each such identifier is an IRI, and is unique to the specification, version, and conformance target.
4. For the conformance target with the name `$conformance-target-name` and the conformance target identifier `$conformance-target-identifier`, the specification includes normative text similar to the following:

A `$conformance-target-name` **MUST** be a *conformant document* as defined by the *NIEM Conformance Targets Attribute Specification* Version 3.0, and **MUST** have an *effective conformance target identifier* of `$conformance-target-identifier`.

A specification that uses the conformance targets attribute as an optional hint would replace the word **MUST** in the above rule with the word **MAY** or **SHOULD**, as appropriate for the level of requirement desired.

Appendix A. References

- [ClarkNS]:** Clark, J. (1999, February 4). *XML Namespaces*. Retrieved June 21, 2012, from <http://www.jclark.com/xml/xmlns.htm>
- [OASISConformance]:** *Guidelines to Writing Conformance Clauses*. (2007, September 4). OASIS Technical Committee Handbook. Retrieved June 15, 2012, from <http://docs.oasis-open.org/templates/TCHandbook/ConformanceGuidelines.html>
- [RFC2119]:** Bradner, S. (1997, March). *Key words for use in RFCs to Indicate Requirement Levels*. Internet Engineering Task Force. Retrieved from <http://www.ietf.org/rfc/rfc2119.txt>
- [RFC3987]:** Duerst, M., & Suignard, M. (2005, January). *Internationalized Resource Identifiers (IRIs)*, RFC 3987. Retrieved June 21, 2012, from <http://www.ietf.org/rfc/rfc3987>
- [XML]:** Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (2008, November 26). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. The World Wide Web Consortium (W3C). Retrieved from <http://www.w3.org/TR/xml/>
- [XMLSchema1]:** Thompson, H. S., Beech, D., Maloney, M., & Mendelsohn, N. (2004, October 28). *XML Schema Part 1: Structures Second Edition*. Retrieved from <http://www.w3.org/TR/xmlschema-1/>.
- [XMLNamespaces]:** Bray, T., Hollander, D., Layman, A., Tobin, R., & Thompson, H. S. (2009, December 8). *Namespaces in XML 1.0 (Third Edition)*. W3C. Retrieved from <http://www.w3.org/TR/2009/REC-xml-names-20091208/>
- [XPathDM]:** Berglund, A., Fernández, M. F., Malhotra, A., Marsh, J., Nagy, M., & Walsh, N. (2010, December 14). *XQuery 1.0 and XPath 2.0 Data Model (XDM) (Second Edition)*. Retrieved June 21, 2012, from <http://www.w3.org/TR/xpath-datamodel/>
- [XSLT1]:** Clark, J. (1999, November 16). *XSL Transformations (XSLT) Version 1.0*. W3C. Retrieved from <http://www.w3.org/TR/xslt>
- [XSLT2]:** Kay, M. (2007, January 23). *XSL Transformations (XSLT) Version 2.0*. W3C. Retrieved from <http://www.w3.org/TR/2007/REC-xslt20-20070123/>

Appendix B. XML Schema Document for Conformance Targets Attribute

Figure B-1: XML Schema Document for Conformance Targets Attribute

```
<?xml version="1.0" encoding="US-ASCII"?>
<xs:schema
  targetNamespace="http://release.niem.gov/niem/conformanceTargets/3.0/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:attribute name="conformanceTargets">
    <xs:simpleType>
      <xs:list itemType="xs:anyURI"/>
    </xs:simpleType>
  </xs:attribute>

</xs:schema>
```

Appendix C. Sample XSLT Version 1.0 Implementation

The following is a sample implementation of XSLT version 1.0 ([XSLT1](#)) templates that interact with the conformance targets attribute. This section is not normative. The key templates are:

- `ctu:get-effective-conformance-targets`: Yields the effective conformance targets for the current XML document.
- `ctu:conformance-targets-contains`: Indicates if the list of conformance targets (passed in as a parameter) contains a specific conformance target IRI.

Figure C-1: Sample XSLT Version 1.0 Implementation

```
<?xml version="1.0" encoding="UTF-8"?>
<stylesheet
  xmlns="http://www.w3.org/1999/XSL/Transform"
  xmlns:ctu="http://example.org/conformance-targets-utilities/xslt1/"
  version="1.0">

  <!--
    Sample use:
    <xsl:call-template name="ctu:get-effective-conformance-targets"/>
    This template yields the effective conformance targets value for the current XML
    document.
  -->
  <template name="ctu:get-effective-conformance-targets">
    <value-of select="//@*[
      namespace-uri() = 'http://release.niem.gov/niem/conformanceTargets/3.0/'
      and local-name() = 'conformanceTargets'
    ] [1]"/>
  </template>

  <!--
    Sample use:

    <xsl:call-template name="ctu:conformance-targets-contains">
      <with-param name="conformance-targets"
        select="/xs:schema/@ct:conformanceTargets"/>
      <with-param name="iri"
        >http://example.org/specification/version/#ConformanceTarget</with-param>
    </xsl:call-template>

    Yields 'true' if the iri appears in the list of conformance targets.
    Yields 'false' otherwise.
  -->
  <template name="ctu:conformance-targets-contains">
    <param name="conformance-targets"/>
    <param name="iri"/>

    <call-template name="ctu:list-contains">
      <with-param name="list" select="$conformance-targets"/>
      <with-param name="member" select="$iri"/>
    </call-template>
  </template>

  <!--
    list operations:
    list is empty: string-length(normalize-space($list)) = 0
  -->

  <!-- Yields the first member of a list. -->
  <template name="ctu:list-get-first">
    <param name="list"/>
    <param name="ns-list" select="normalize-space($list)"/>
    <choose>
      <when test="string-length($ns-list) = 0">
        <message terminate="yes">ERROR in template <!--
          -->{http://example.org/conformance-targets-utilities/xslt1}get-first-from-list: <!--
          -->You can't get the first entry of an empty list.</message>

```


Standards Track Work Product

```
</when>
<when test="contains($ns-list, ' ')">
  <value-of select="substring-before($ns-list, ' ')" />
</when>
<otherwise>
  <value-of select="$ns-list" />
</otherwise>
</choose>
</template>

<!-- Yields everything after the first member of a list. -->
<template name="ctu:list-get-rest">
  <param name="list" />
  <param name="ns-list" select="normalize-space($list)" />
  <choose>
    <when test="string-length($ns-list) = 0">
      <message terminate="yes">ERROR in template <!--
        -->{http://example.org/conformance-targets-utilities/xslt1/}get-first-from-list: <!--
        -->You can't get the rest of an empty list.</message>
    </when>
    <when test="contains($ns-list, ' ')">
      <value-of select="substring-after($ns-list, ' ')" />
    </when>
    <!-- otherwise yield nothing -->
  </choose>
</template>

<!-- Yields true iff the list contains the member. -->
<template name="ctu:list-contains">
  <param name="list" />
  <param name="member" />
  <choose>
    <when test="string-length(normalize-space($list)) = 0">
      <value-of select="false()" />
    </when>
    <otherwise>
      <variable name="first">
        <call-template name="ctu:list-get-first">
          <with-param name="list" select="$list" />
        </call-template>
      </variable>
      <choose>
        <when test="$first = $member">
          <value-of select="true()" />
        </when>
        <otherwise>
          <variable name="rest">
            <call-template name="ctu:list-get-rest">
              <with-param name="list" select="$list" />
            </call-template>
          </variable>
          <choose>
            <when test="string-length($rest) = 0">
              <value-of select="false()" />
            </when>
            <otherwise>
              <call-template name="ctu:list-contains">
                <with-param name="list" select="$rest" />
                <with-param name="member" select="$member" />
              </call-template>
            </otherwise>
          </choose>
        </otherwise>
      </choose>
    </otherwise>
  </choose>
</template>
</stylesheet>
```

Appendix D. Sample XSLT Version 2.0 Implementation

The following is a sample implementation of XSLT version 2.0 ([XSLT2](#)) functions that interact with the conformance targets attribute. This section is not normative. The key functions are:

- `ctu:get-effective-conformance-targets($node as node()) as xs:anyURI*`: Yields the effective conformance targets for the XML document containing the given node.
- `ctu:conformance-targets-contains($conformance-targets as xs:anyURI*, $iri as xs:anyURI) as xs:boolean`: Indicates if the given sequence of conformance targets contains the given IRI.

Figure D-1: Sample XSLT Version 2.0 Implementation

```
<?xml version="1.0" encoding="UTF-8"?>
<stylesheet
  xmlns="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ctu="http://example.org/conformance-targets-utilities/xslt2/"
  version="2.0">

  <!--
  Sample use:
  <xsl:sequence select="
    ctu:get-effective-conformance-targets(/)
  "/>
  This function yields the effective conformance targets value for XML document.
  -->
  <function name="ctu:get-effective-conformance-targets" as="xs:anyURI*">
    <param name="node" as="node()" />
    <sequence select="for $string in tokenize(normalize-space(
      (root($node)//@*[
        namespace-uri() = 'http://release.niem.gov/niem/conformanceTargets/3.0/'
        and local-name() = 'conformanceTargets'
      ])[1]), ' ') return xs:anyURI($string)"/>
  </function>

  <!--
  Sample use:
  <xsl:if test="
    ctu:conformance-targets-contains(
      ctu:get-effective-conformance-targets(/),
      xs:anyURI('http://example.org/specification/version/#ConformanceTarget')
    )">
  This function yields true if and only if the list of conformance targets contains the
  given iri.
  -->
  <function name="ctu:conformance-targets-contains" as="xs:boolean">
    <param name="conformance-targets" as="xs:anyURI*" />
    <param name="iri" as="xs:anyURI" />

    <sequence select="$iri = $conformance-targets"/>
  </function>

</stylesheet>
```

Appendix E. Index of definitions

- [conformance target: Section 3.1, *Conformance Target Defined*](#)
- [conformance target identifier: Section 3.1, *Conformance Target Defined*](#)
- [conformant document: Section 3.2, *Conformance to this Specification*](#)
- [effective conformance target identifier: Section 4, *Semantics and Use*](#)
- [effective conformance targets attribute: Section 4, *Semantics and Use*](#)

Appendix F. Index of rules

- [Rule 3-1, Conformant document is an XML document: Section 3.3, Syntactic Conformance](#)
- [Rule 3-2, Attribute `ct:conformanceTargets` is valid: Section 3.3, Syntactic Conformance](#)
- [Rule 3-3, Namespace defines only attribute `ct:conformanceTargets`: Section 3.3, Syntactic Conformance](#)
- [Rule 3-4, No reference to types in conformance targets namespace: Section 3.3, Syntactic Conformance](#)
- [Rule 3-5, Attribute `ct:conformanceTargets` contains only absolute IRI references: Section 3.3, Syntactic Conformance](#)
- [Rule 3-6, Document may contain any number of `ct:conformanceTargets` attributes: Section 3.3, Syntactic Conformance](#)

Appendix G. Acknowledgments

Special Thanks

The NIEM Technical Architecture Committee (NTAC) team members wish to express their gratitude to Webb Roberts for his contributions.

Participants

The following individuals are active members of the NTAC at the time of this publication:

- Aubrey Beach
- Jim Cabral
- Tom Carlson
- Mike Douklias
- Katherine Escobar
- Mike Hulme
- Eric Jahn
- Dave Kemp
- Vamsi Kondannagari
- Peter Madruga
- Christina Medlin
- Joe Mierwa
- Scott Renner
- Beth Smalley
- Duncan Sparrell
- Jennifer Stathakis
- Stephen Sullivan

Appendix H. Notices

Copyright © OASIS Open 2023. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This specification is published under [Attribution 4.0 International \(CC BY 4.0\)](#). Code associated with this specification is provided under [Apache License 2.0](#).

All contributions made to this project have been made under the [OASIS Contributor License Agreement \(CLA\)](#).

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the [NIEMOpen IPR Statement](#) page.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Open Project (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards Final Deliverable documents (Project Specifications, OASIS Standards, or Approved Errata).

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Open Project that produced this deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Open Project that produced this OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Open Project can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.]

Standards Track Work Product

The name "OASIS" is a trademark of [OASIS](https://www.oasis-open.org/policies-guidelines/trademark/), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark/> for above guidance.