

OGC® DOCUMENT: 24-066R1

External identifier of this OGC® document: <http://www.opengis.net/doc/POL/ModSpec-part3/1.1>



Open
Geospatial
Consortium

THE MODSPEC MODEL - PART 3: XML

STANDARD
Implementation

APPROVED

Version: 1.1

Submission Date: 2025-10-20

Approval Date: 2026-03-05

Publication Date: 2026-05-21

Editor: Carl Reed, John Herring

Notice: This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: <http://ogc.standardstracker.org/>

Copyright notice

Copyright © 2026 Open Geospatial Consortium

To obtain additional rights of use, visit <https://www.ogc.org/legal>

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I. KEYWORDS	v
II. PREFACE	vi
III. SECURITY CONSIDERATIONS	vii
IV. SUBMITTING ORGANIZATIONS	viii
V. DOCUMENT TERMS AND DEFINITIONS	viii
VI. DOCUMENT EDITORS	viii
VII. DOCUMENT CONTRIBUTORS	ix
VIII. SUBMITTING ORGANIZATIONS	ix
IX. REVISION HISTORY	x
X. FOREWORD	x
1. SCOPE	2
1.1. Introduction	2
2. CONFORMANCE	4
3. NORMATIVE REFERENCES	6
4. TERMS AND DEFINITIONS	8
6. CONVENTIONS	10
6.1. Symbols (and abbreviated terms)	10
6.2. Identifiers	10
6.3. Abbreviated terms	10
6.4. Finding requirements and recommendations	11
7. REQUIREMENTS CLASS: PART 2: XML	13
7.1. The ModSpec and the “Form” of a standard	13
8. CONFORMANCE TEST CLASS: XML SCHEMA	21
8.1. Dependency on Core	21
8.2. Dependency on W3C XML schema	21

8.3. A requirements class corresponds to a single XML namespace	21
8.4. A reference to the URI of the test suite in ApplInfo	22
8.5. Direct dependencies become schema imports	22
8.6. No requirements class modifies or redefines elements in another namespace	23
9. CONFORMANCE TEST CLASS: SCHEMATRON	25
9.1. Dependency on XML Schema conformance test class	25
9.2. Each schematron pattern element implements one requirement	25
9.3. Each schematron pattern is in one schematron element	25
9.4. Each schematron @fpi attribute is used only once	26
9.5. Each schematron @see attribute is used only once	26
9.6. Each schematron fpi attribute is used only once	27
10. CONFORMANCE CLASS: XML META-SCHEMA	29
10.1. Supports the Specification class	29
10.2. Each XML schema is conformant with the XML Schema class	29



KEYWORDS

The following are keywords to be used by search engines and document catalogues.

oogcdoc, OGC document, modspec, modular, requirements, testing, part3, xml



PREFACE

This Open Geospatial Consortium (OGC) member developed and approved document, referred to as the ModSpec: Part 3, defines an optional requirements class for the XML model extension the ModSpec core.

The ModSpec: Core Standard defines a model and related requirements and recommendations for writing and structuring modular standards documents. Further, this model is designed to enable consistent and verifiable testing of implementations of a standard that claim conformance. The ModSpec is a meta-standard: A standard specifying requirements for crafting and documenting modular and testable standards.

The goals of using the ModSpec are:

- To define characteristics and a structure for the development of modular standards which will minimize the difficulty in writing testable standards while maximizing usability and interoperability; and
- To ensure that a standard specifies requirements in a common and consistent manner and that these requirements are testable.

Suggested additions, changes, and comments on this document are welcome and encouraged. Such suggestions may be submitted by creating an issue in the GitHub repository for this document (<https://github.com/opengeospatial/ogc-modspec>).



SECURITY CONSIDERATIONS

No security considerations have been made for this document.

IV

SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Carl Reed Associates
- Heazeltech
- Tapa Enterprise Solutions (ImageMatters)
- The Danish Agency for Climate Data
- UK Met Office

V

DOCUMENT TERMS AND DEFINITIONS

This document uses the standard terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the imperative verb form used to indicate a requirement to be strictly followed to conform to this standard.

VI

DOCUMENT EDITORS

The following OGC Members participated in editing this document:

PERSON	ORGANIZATION REPRESENTED
Carl Reed	Carl Reed & Associates
John Herring, PhD	Oracle

VII

DOCUMENT CONTRIBUTORS

The following OGC Members contributed and participated in developing Version 1.1 of the ModSpec.

PERSON	ORGANIZATION REPRESENTED
Carl Reed, PhD	Carl Reed
Chuck Heazel	Heazeltech
Jeff Yutzler	ImageMatters

VIII

SUBMITTING ORGANIZATIONS

The following OGC Members support the ModSpec Version 1.1 submission

ORGANIZATION
Carl Reed & Associates
Heazeltech
ImageMatters/Tema
UK Met Office



REVISION HISTORY

This is the second normative version of this document.



FOREWORD

The ModSpec specifies a formal structure for standards documents but does not supply specific content. This Part 3 specifies the optional XML Requirements Class.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

1

SCOPE

1

SCOPE

The OGC ModSpec Standard defines characteristics and structure for the specification of Standards that will encourage implementation by minimizing difficulty determining requirements, mimicking implementation structure and maximizing usability and interoperability. Part 3 of the ModSpec defines the requirements for using XML. The XML related requirements classes cover any standard which has as one of its purposes the introduction of a new XML schema. Such a standard would normally define the schema, all of its components, and its intended uses.

1.1. Introduction

1.1.1. Semantics

The ModSpec Core Standard defines requirements for conformance to the ModSpec, but testing for that conformance may depend on how the various forms and parts of a conformant standard are viewed. The ModSpec Part 3 Standard specifies how those views are to be defined as XML.

As suggested in Clause Requirement 8 of the ModSpec Core, the structure of the normative clauses in a standard should parallel the structure of the conformance test classes in that standard. The structure of the normative clauses in a well written standard will follow the structure of its model. This means that all three are parallel.

1.1.2. ModSpec document structure

Version 1.1 of the ModSpec is split into a Core standard and multiple Parts. These are:

- Part 1: Core – contains all the core requirements and informational text that define the model and internal structure of a standard;
- Part 2 – UML Model requirements; and
- Part 3 – XML Model and Schematron requirements.

2

CONFORMANCE

2

CONFORMANCE

Conformance to the ModSpec Part 3: XML by technical implementation standards can be tested by inspection. The test suite is in [annex-A].

There are two conformance classes in this document for:

1. Standards using XML to state requirements, extending the core; and
2. Standards using Schematron. This class covers any standard that uses Schematron to create patterns or constraints for an XML Schema.

This document contains normative language and thus places requirements on conformance, or mechanism for adoption, of candidate standards to which the ModSpec applies. In particular:

- The OGC ModSpec: Core specifies the core requirements which *SHALL* be met by all conformant standards; and
- The various subclauses of Clause 7 list requirements partially derived from the core, but more specific to the conditions where the data model is expressed in XML. The XML requirements class is an extension of the Core Requirements Class presented in the ModSpec Core Standard.



3

NORMATIVE REFERENCES

3

NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

W3C: W3C xmlschema-1, *XML Schema Part 1: Structures Second Edition*. World Wide Web Consortium <https://www.w3.org/TR/xmlschema-1/>.

W3C: W3C xmlschema-2, *XML Schema Part 2: Datatypes Second Edition*. World Wide Web Consortium <https://www.w3.org/TR/xmlschema-2/>.

International Organization for Standardization (committee): ISO/IEC 19757-3:2006, *Information technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation – Schematron*. International Organization for Standardization, International Electrotechnical Commission, Geneva (2006). <https://www.iso.org/standard/40833.html>.

4

TERMS AND DEFINITIONS

4

TERMS AND DEFINITIONS

No terms and definitions are listed in this document.

All terms and definitions in Clause 4 of the OGC ModSpec: Core are relevant and used in this Part.

Terms not defined here take their meaning from computer science or from their Standard English (common US and UK) meanings.



6

CONVENTIONS

6

CONVENTIONS

6.1. Symbols (and abbreviated terms)

All symbols used in this document are either:

1. Common mathematical symbols; or
2. UML 2 (Unified Modeling Language) as defined by OMG and accepted as a publicly available standard (PAS) by ISO in its earlier 1.3 version.

6.2. Identifiers

The normative provisions in this standard are denoted by the URI namespace

<https://www.opengis.net/spec/modspec-3/1.1/>

All requirements that appear in this document are denoted by partial URIs which are relative to the namespace shown above.

For the sake of brevity, the use of “req” in a requirement URI denotes <https://www.opengis.net/spec/modspec-3/1.1/>

An example might be:

`/req/part3/xml`

All conformance tests that appear in this document are denoted by partial URIs which are relative to the namespace shown above.

For the sake of brevity, the use of “conf” in a requirement URI denotes:

<https://www.opengis.net/spec/modspec-3/1.1/>

The same convention is used for permissions (per) and recommendations (rec).

6.3. Abbreviated terms

This document uses the same abbreviations and acronyms as defined in the ModSpec Core Standard

6.4. Finding requirements and recommendations

Each normative statement in the ModSpec Part 3 Standard is stated in one and only one place, in a standard format, and with a unique label, such as REQ001, REC001, or PER001. A requirement, recommendation, or permission may be repeated for clarification. The statement with the unique label is considered the official statement of the normative requirement or recommendation.

In this document, all requirements are associated with tests specified in the test suite in [annex-A]. The reference to the requirement in the test case is done by a requirements label. Recommendations are not tested although they still are documented using a standard template and have unique identifiers.

Requirements classes are separated into their own clauses and named, and specified according to inheritance (direct dependencies). The Conformance test classes in the test suite are similarly named to establish an explicit link between requirements classes and conformance test classes.

7

REQUIREMENTS CLASS: PART 2: XML

7

REQUIREMENTS CLASS: PART 2: XML

This clause defines the key concepts and requirements that represent Part 3 XML schema extension to the ModSpec Core.

7.1. The ModSpec and the “Form” of a standard

NOTE: For OGC Standards, the assumption is that documents are in a commonly used logical form (template).

In informative sections, the use of the word “will” implies that something is an implication of a requirement. The “will” statements are not requirements but explain the consequence of requirements.

The ModSpec defines a “requirement” of a standard as an testable criterion. See the formal definition of requirement in the ModSpec Core document.

7.1.1. Requirements class: The XML schema extension to the core

The XML schema extension requirements class covers any standard which has as one of its purposes the introduction of a new XML schema. Such a standard would normally define the schema, all of its components, and its intended uses.

Table 1

Requirements Class – XML extension to the core	
/req/core/data-representation	
Target	ModSpec Conformant XML Model
Dependency	OGC ModSpec Version 1.1 [25-003r1]
REQ001	/req/part3/xml/conformance-with-core
REQ002	/req/part3/xml/w3c-xml
REQ003	/req/part3/xml/single-xml-space

REQ004	/req/part3/xml/all-components-uri
REQ005	/req/part3/xml/direct-dependency
REQ006	/req/part3/xml/no-element-modification

First, by the requirements above, the extension relationship of this conformance test class to the core must be made explicit.

Table 2

REQ001	/req/part3/xml/conformance-with-core An implementation passing the XML conformance test class <i>SHALL</i> first pass the ModSpec Core conformance test class
--------	---

Table 3

REQ002	/req/part3/xml/w3c-xml An implementation passing the XML schema conformance test class <i>SHALL</i> first pass the W3C Recommendation for XML schema.
--------	---

Each XML schema file (usually *.xsd) carries a target namespace specification, recorded in the `targetNamespace` attribute of the `<schema>` element in the XML representation. In its implementation, this namespace is represented by a globally unique identifier, a URI. All schema components defined with that URI as its namespace designation are part of the same module in XML schema.

The XML Schema specification lists those resolution strategies for namespace and schema that a schema-aware process may use. They work in a predictable fashion independent of the choice of strategy if and only if the schemas are in a one to one correspondence to their namespace. A schema may be dependent on another schema and may contain “import” directives that load all such schemas whenever it is loaded.

When a process loads a schema as defined by its namespace URI identity, it must always get a linkage to all components in that namespace. If not, then at sometime in the future, the process will fail when it finds a reference to such a component that it missed. To prevent this sort of failure, when a schema-aware process first encounters a namespace URI it must always be associated to a schema location (a file) that contains or includes all schema components having the URI as their namespace. This is referred to as the “all-components schema document”.

In defining a XML schema (either completely, or partially in a standard) the fundamental component or module of XML schema is always the namespace and its associated schema, which is designated by a URI.

Table 4

REQ003	<p>/req/part3/xml/single-xml-space</p> <p>If a standard conformant to the XML schema conformance class defines a set of data schemas, all components (e.g., elements, attributes, types ...) associated with a single conformance test class <i>SHALL</i> be scoped to a single XML namespace.</p>
--------	---

Table 5

REQ004	<p>/req/part3/xml/all-components-uri</p> <p>The all-components schema document for an XML Schema <i>SHALL</i> indicate the URI of the associated conformance test class in the schema/annotation/appinfo element.</p>
--------	--

The mechanism for dependencies may either be by importation or by inclusion of schema components.

Example: In GML 3, the spatial schema (ISO 19107) and the general feature model (ISO 19109) are both satisfied by elements within the single GML namespace. A viable alternative would to have put the schema components for spatial schema and feature schema in separate namespaces.

This is a choice of design, and at the level of the ModSpec, the trade-off factors cannot be prejudged because the details of such cost-benefit trade-offs are not constant. Either of the above approaches may be used.

Table 6

REQ005	<p>/req/part3/xml/direct dependency</p> <p>If a standard conformant to the XML schema conformance class defines a direct dependency from one requirement class to another, then a standardization target of the corresponding conformance test class <i>SHALL</i> import a schema that has passed the associated conformance test class (dependency) or shall itself pass the associated conformance test class.</p>
--------	---

NOTE: This implies that the value of the schemaLocation on the <import> element will refer to the all-components schema document.

Table 7

PER001	<p>/per/part3/xml/schema-assembly</p> <p>An all-components schema document <i>MAY</i> be assembled by inclusion of documents that describe subsets of the components associated with the conformance test class.</p>
--------	---

This allows schema designers to do some modularization within a namespace for convenience, notwithstanding the requirement for an all-components schema document.

NOTE: A namespace variable is used if the requirements class is not defining a schema, but defining requirements for a schema to be the target of its conformance class. For example, GML defines requirements for application schemas, but does not a priori know the namespace of any application schema. The namespace for the application schema becomes a variable in the conformance test cases.

Table 8

REQ006	/req/part3/xml/no-element-modification The all-components schema document for an XML Schema <i>SHALL</i> indicate the URI of the associated conformance test class in the schema/annotation/appinfo element.
---------------	--

Table 9

PER002	/per/part3/xml/add-constraints Requirements may add constraints.
---------------	--

This allows extensions to restrict the following.

1. Usage of existing elements, but only if their use was originally optional. This is similar to the rules for inheritance (such as in UML or other object models), where a class can eliminate an attribute from a superclass only if the superclass attribute includes a “0” in its multiplicity range.
2. The type of existing elements, to sub-types of the original elements. This is similar to the rules for inheritance, where a class can re-define an attribute or association role from a superclass so that its type or class is a specialization of the original.

In summary, effective modularization is enabled by following the pattern of one conformance class per XML namespace; i.e., the set of components in an XML namespace should be referred to as a whole. Subsetting of components in a single XML namespace for conformance purposes is not permitted.

7.1.2. Optional Requirements class: Schematron extension

Schematron (ISO/IEC 19757-3:2006) provides a notation with which many constraints on XML documents can be expressed. As such, Schematron is a rule-based validation language for making assertions about the presence or absence of patterns in XML trees. It is a structural schema language expressed in XML using a small number of elements and XPath languages.

This requirements class covers any standard that uses Schematron to create patterns or constraints for an XML Schema.

Table 10

Requirements Class – XML Schematron extension to the core	
/req/core/data-representation	
Target	ModSpec Conformant XML Model
Dependency	OGC ModSpec Version 1.1 (need proper title and document number)
REQ007	/req/part3/xml/schematron-xml-schema
REQ008	/req/part3/xml/sch-pattern-constraints
REQ010	/req/part3/xml/fpi-attribute-is-uri
REQ011	/req/part3/xml/see-attribute-is-identifier
REQ012	/req/part3/xml/one-fpi-attribute-per-schema

First the schema must be defined within the bounds of the XML schema requirements class.

Table 11

REQ007	<p>/req/part3/xml/schematron-xml-schema</p> <p>A standard passing the Schematron conformance test class <i>SHALL</i> also define or reference an XML schema that shall pass the XML schema conformance class from this standard.</p>
--------	---

Within a Schematron schema, the “pattern” and “schema” elements may be used in a way that corresponds with conformance tests and a conformance test class as follows:

Table 12

REQ008	/req/part3/xml/sch-pattern-constraints
A	Each sch:assert element <i>SHALL</i> implement constraints described in no more than one requirement.
B	Each requirement <i>SHALL</i> be implemented by no more than one sch:pattern.

NOTE: Requirement 9 was removed in the ModSpec version 1.1 as this requirement is present in the latest Schematron RelaxNG schema.

The Formal Public Identifier (fpi) attribute names the Schematron schema. In the ModSpec Part 2 extension, this identifier is a URI.

Table 13

	/req/part3/xml/fpi-attribute-is-uri
REQ010	The value of the sch:schema/@fpi attribute <i>SHALL</i> be a URI that identifies this implementation.

The @see attribute in Schematron provides a hypertext link to documentation or related material for each pattern, rule, or assertion. This allows the Schematron schema to be integrated into a wider information system. In the ModSpec Part 2 extension, this attribute is the identifier for a requirements class.

Table 14

	/req/part3/xml/see-attribute-is-identifier
REQ011	The value of the sch:schema/@see attribute <i>SHALL</i> be the identifier for the requirements class that contains the requirement(s) implemented by the schema

Table 15

	/req/part3/xml/one-fpi-attribute-per-schema
REQ012	The value of the sch:schema/@fpi attribute <i>SHALL</i> be used on only one Schematron schema.

7.1.3. Optional Requirements class: XML meta-schema extension to the ModSpec Core.

This requirements class covers any standard which has as one of its purposes the introduction of a new type of XML schema. Such a standard would normally define the characteristics of such schema, how its components are created and its intended uses.

First, by the requirements above, the extension relationship of this conformance test class to the core must be made explicit.

Table 16

	/req/part3/xml/first-pass-core-conformance
REQ013	A standard passing the XML meta-schema conformance test class <i>SHALL</i> first pass the ModSpec Core conformance test class.

Since the target is defining requirements for XML schemas, it will require that the ModSpec be used.

Table 17

REQ014	/req/part3/xml/pass-XML-schema-conformance-class A standard passing the XML meta-schema conformance test class <i>SHALL</i> require that its targets (XML schema) pass the XML schema conformance class.
--------	--

8

CONFORMANCE TEST CLASS: XML SCHEMA

8

CONFORMANCE TEST CLASS: XML SCHEMA

8.1. Dependency on Core

An implementation passing the XML schema conformance test class shall first pass the ModSpec Core Standard conformance test class.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 2
4. Test Type: Conformance.

8.2. Dependency on W3C XML schema

An implementation passing the XML schema conformance test class shall first pass the W3C Recommendation for XML schema.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 3
4. Test Type: Conformance.

8.3. A requirements class corresponds to a single XML namespace

If a standard conformant to the XML schema conformance class defines a set of data schemas, all components (e.g., elements, attributes, types ...) associated with a single conformance test class shall be scoped to a single XML namespace.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 4
4. Test Type: Conformance.

8.4. A reference to the URI of the test suite in ApplInfo

The all-components schema document for an XML Schema shall indicate the URI of the associated conformance test class in the schema/annotation/appinfo element.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 5
4. Test Type: Conformance.

8.5. Direct dependencies become schema imports

If a standard conformant to the XML schema conformance class defines a direct dependency from one requirement class to another, then a standardization target of the corresponding conformance test class shall import a schema that has passed the associated conformance test class (dependency) or shall itself pass the associated conformance test class.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.

3. Reference: Table 6
4. Test Type: Conformance.

8.6. No requirements class modifies or redefines elements in another namespace

No requirements class in a standard conformant to the XML schema conformance class shall modify elements, types or any other requirement from a namespace to which it is not associated.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 8
4. Test Type: Conformance.

9

CONFORMANCE TEST CLASS: SCHEMATRON

9.1. Dependency on XML Schema conformance test class

A standard passing the Schematron conformance test class shall also define or reference an XML schema that shall pass the XML schema conformance class from this standard.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 11
4. Test Type: Conformance.

9.2. Each schematron pattern element implements one requirement

Each sch:pattern element shall implement constraints described in no more than one requirement. Each requirement shall be implemented by no more than one sch:pattern.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 12
4. Test Type: Conformance.

9.3. Each schematron pattern is in one schematron element

Each sch:pattern element shall be contained within one sch:schema element.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: [req-09]
4. Test Type: Conformance.

9.4. Each schematron @fpi attribute is used only once

The value of the sch:schema/@fpi attribute shall be used on only one Schematron schema.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 13
4. Test Type: Conformance.

9.5. Each schematron @see attribute is used only once

The value of the sch:schema/@see attribute shall be the identifier for the requirements class that contains the requirement(s) implemented by the schema

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 14
4. Test Type: Conformance.

9.6. Each schematron fpi attribute is used only once

The value of the sch:schema/@fpi attribute shall be used on only one Schematron schema.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 15
4. Test Type: Conformance.

10

CONFORMANCE CLASS: XML META-SCHEMA

10.1. Supports the Specification class

A standard passing the XML meta-schema conformance test class shall first pass the ModSpec Core conformance test class.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 16
4. Test Type: Conformance.

10.2. Each XML schema is conformant with the XML Schema class

A standard passing the XML meta-schema conformance test class shall require that its standardization targets (XML schema) pass the XML schema conformance class from this standard.

1. Test Purpose: Verify that this requirement is satisfied.
2. Test Method: Inspect the document to verify the above.
3. Reference: Table 17
4. Test Type: Conformance.