



BSI Standards Publication

Information technology — Document Schema Definition Languages (DSDL)

Part 3: Rule-based validation using Schematron

National foreword

This British Standard is the UK implementation of ISO/IEC 19757-3:2025. It supersedes BS ISO/IEC 19757-3:2020, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/41, Document description and processing language.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 24486 1
ICS 35.240.30

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2025.

Amendments/corrigenda issued since publication

Date	Text affected



**International
Standard**

ISO/IEC 19757-3

**Information technology —
Document Schema Definition
Languages (DSDL) —**

**Part 3:
Rule-based validation using
Schematron**

*Technologies de l'information — Langages de définition de
schéma de documents (DSDL) —*

Partie 3: Validation basée sur des règles à l'aide de Schematron

**Fourth edition
2025-09**

Reference number
ISO/IEC 19757-3:2025(en)

Full document details: BS ISO/IEC 19757-3:2025


© ISO/IEC 2025

Documents
Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

	Page
Foreword.....	v
Introduction.....	vi
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions.....	2
4 Notation.....	4
5 Syntax.....	4
5.1 Well-formedness.....	4
5.2 Namespace.....	4
5.3 Whitespace.....	4
5.4 Core elements.....	5
5.4.1 General.....	5
5.4.2 active element.....	5
5.4.3 assert element.....	5
5.4.4 extends element.....	5
5.4.5 group element.....	6
5.4.6 include element.....	6
5.4.7 let element.....	6
5.4.8 library element.....	6
5.4.9 name element.....	6
5.4.10 ns element.....	6
5.4.11 param element.....	7
5.4.12 pattern element.....	7
5.4.13 phase element.....	7
5.4.14 report element.....	8
5.4.15 rule element.....	8
5.4.16 rules element.....	8
5.4.17 schema element.....	8
5.4.18 value-of element.....	9
5.5 Ancillary elements and attributes.....	9
5.5.1 diagnostic element.....	9
5.5.2 diagnostics attribute.....	9
5.5.3 diagnostics element.....	9
5.5.4 dir element.....	9
5.5.5 emph element.....	9
5.5.6 flag attribute.....	9
5.5.7 from attribute.....	10
5.5.8 fpi attribute.....	10
5.5.9 icon attribute.....	10
5.5.10 p element.....	10
5.5.11 properties attribute.....	10
5.5.12 properties element.....	10
5.5.13 property element.....	10
5.5.14 role attribute.....	11
5.5.15 schematronEdition attribute.....	11
5.5.16 severity attribute.....	11
5.5.17 see attribute.....	11
5.5.18 span element.....	11
5.5.19 subject attribute.....	11
5.5.20 title element.....	12
5.5.21 visit-each attribute.....	12
5.5.22 when attribute.....	12
6 Semantics.....	12

6.1	Validation function	12
6.2	Minimal syntax.....	13
6.3	Processing model	13
6.4	Abstract pattern processing.....	14
6.5	Query language binding.....	16
6.6	Order and side-effects.....	16
6.7	Localization and language fixup.....	17
6.8	Base URI fixup.....	17
7	Conformance	17
7.1	Simple conformance	17
7.2	Full conformance.....	17
Annex A (normative) RELAX NG schema for Schematron		19
Annex B (normative) Schematron schema for additional constraints.....		23
Annex C (normative) Default query language binding		25
Annex D (informative) Schematron Validation Report Language		26
Annex E (informative) Design requirements		35
Annex F (informative) Use of Schematron as a vocabulary.....		36
Annex G (informative) Use of Schematron for multi-lingual schemas		37
Annex H (normative) Query language binding for XSLT2.....		38
Annex I (normative) Query language binding for XPath2		40
Annex J (normative) Query language binding for XSLT3		41
Annex K (normative) Query language binding for XPath3		43
Annex L (normative) Query language binding for XPath31		44
Annex M (normative) Query language binding for XQuery3		45
Annex N (normative) Query language binding for XQuery31.....		47
Annex O (informative) Example usage of Schematron properties.....		49
Bibliography		51

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see <https://www.iso.org/directives> or https://www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at <https://www.iso.org/patents> and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see <https://www.iso.org/iso/foreword.html>. In the IEC, see <https://www.iec.ch/understanding-standards>.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*.

This fourth edition cancels and replaces the third edition (ISO/IEC 19757-3:2020 [\[1\]](#)), which has been technically revised.

The main changes are as follows:

- a query language binding has been added for XQuery3 ([Annex M](#));
- new elements `group` ([5.4.5](#)), `library` ([5.4.8](#)) and `rules` ([5.4.16](#)) have been added;
- new attribute `as` has been added to element `let` ([5.4.7](#)), to enable datatyping of variables;
- new attributes `schematronEdition` ([5.5.15](#)) and `severity` ([5.5.16](#)) have been added;
- `param` elements ([5.4.11](#)) are now allowed for schemas and abstract patterns and in the latter case their value can act as a default;
- attributes `flag` ([5.5.6](#)), `role` ([5.5.14](#)) and `severity` ([5.5.16](#)) are dynamically evaluated if their value is a variable reference;
- element `extends` ([5.4.4](#)) is made available at the top level of a schema;
- the methods of base URI fixup ([6.8](#)) and language fixup ([6.7](#)) are specified;
- a mapping of schema to SVRL structures is provided in [Annex D](#).

A list of all parts in the ISO/IEC 19757 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

ISO/IEC 19757 (all parts) [\[2\]](#) defines a set of Document Schema Definition Languages (DSDL) that can be used to specify one or more validation processes performed against Extensible Markup Language (XML 1.0 [\[3\]](#)) or Standard Generalized Markup Language (SGML) documents. [XML is an application profile SGML (see ISO 8879 [\[4\]](#)).

A document model is an expression of the constraints to be placed on the structure and content of documents to be validated with the model. A number of technologies have been developed through various formal and informal consortia since the development of Document Type Definitions (DTDs) as part of ISO 8879 [\[4\]](#), notably by the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). A number of validation technologies are standardized in DSDL to complement those already available as standards or from the industry.

Through the validation that a structured document conforms to specified constraints in structure and content, the potentially many applications acting on the document are relieved from duplicating the task of confirming that such requirements have been met. Historically, such tasks and expressions have been developed and utilized in isolation, without consideration of how the features and functionality available in other technologies can enhance validation objectives.

The main objective of ISO/IEC 19757 (all parts) [\[2\]](#) is to bring together different validation-related tasks and expressions to form a single extensible framework that allows technologies to work in series or in parallel to produce a single or a set of validation results. The extensibility of DSDL accommodates validation technologies not yet designed or specified.

In the past, different design and use criteria have led users to choose different validation technologies for different portions of their information. Bringing together information within a single XML document sometimes prevents existing document models from being used to validate sections of data. By providing an integrated suite of constraint description languages that can be applied to different subsets of a single XML document, ISO/IEC 19757 (all parts) [\[2\]](#) allows different validation technologies to be integrated under a well-defined validation policy.

The structure of this document is as follows. [Clause 5](#) describes the syntax of a Schematron schema. [Clause 6](#) describes the semantics of a correct Schematron schema; the semantics specify when a document is valid with respect to a Schematron schema. [Clause 7](#) describes conformance requirements for implementations of Schematron validators. [Annex A](#) provides the ISO/IEC 19757-2 [\[5\]](#) (RELAX NG) schema for Schematron. [Annex B](#) provides the Schematron schema for constraints in Schematron that cannot be expressed by the schema of [Annex A](#). [Annex C](#) provides the default query language binding to XSLT1. [Annex D](#) provides an ISO/IEC 19757-2 [\[5\]](#) (RELAX NG compact syntax) schema and corresponding Schematron schema for a simple XML language Schematron Validation Report Language. [Annex E](#) provides motivating design requirements for Schematron. [Annex F](#) specifies certain Schematron elements to be used in external vocabularies. [Annex G](#) provides a simple example of a multi-lingual schema. [Annex H](#) to [Annex M](#) provide query language bindings. [Annex O](#) shows example usage of Schematron properties.

This edition is backwards compatible with ISO/IEC 19757-3:2020 [\[1\]](#) to the extent that validation results produced by existing schemas are unaffected by the changes this edition makes, supersedes it and provides extra query language bindings, for XQuery3.

Considered as a document type, a Schematron schema contains natural-language assertions concerning a set of documents, marked up with various elements and attributes for testing these natural-language assertions and for simplifying and grouping assertions.

Considered theoretically, a Schematron schema reduces to a non-chaining rule system whose terms are Boolean functions invoking an external query language on the instance and other visible XML documents, with syntactic features to reduce specification size and to allow efficient implementation.

Considered analytically, Schematron has two characteristic high-level abstractions: the pattern and the phase. These allow the representation of non-regular, non-sequential constraints that ISO/IEC 19757-2 [\[5\]](#) cannot specify and various dynamic or contingent constraints.

ISO/IEC 19757-3:2025(en)

This document is based on the Schematron^[6] assertion language. The `let` element is based on XCSL^[7]. Other features arise from the half-dozen early open-source implementations of Schematron in diverse programming languages and from discussions in electronic forums by Schematron users and implementers.

Information technology — Document Schema Definition Languages (DSDL) —

Part 3: Rule-based validation using Schematron

1 Scope

This document specifies Schematron, a schema language for XML. This document establishes requirements for Schematron schemas and specifies when an XML document matches the patterns specified by a Schematron schema. Schematron uses query languages such as XPath for writing assertions.

2 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.

Extensible Markup Language (XML) 1.0 (Fifth Edition), 26 November 2008, <https://www.w3.org/TR/2008/REC-xml-20081126/>

ISO/IEC 19757-2, *Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG*

XML 1.1, *Extensible Markup Language (XML) 1.1 (Second Edition)*, 16 August 2006, <https://www.w3.org/TR/2006/REC-xml11-20060816/>

XML Inclusions (XInclude) Version 1.0 (Second Edition), 15 November 2006, <https://www.w3.org/TR/2006/REC-xinclude-20061115/>

XML Path Language (XPath) 2.0, W3C Recommendation, 23 January 2007, <https://www.w3.org/TR/xpath20/>

XML Path Language (XPath) 3.0, W3C Recommendation, 8 April 2014, <https://www.w3.org/TR/xpath-30/>

XML Path Language (XPath) 3.1, 21 March 2017, <https://www.w3.org/TR/xpath-31/>

XML Path Language (XPath) Version 1.0, W3C Recommendation, 16 November 1999, <https://www.w3.org/TR/xpath-10/>

XPath and XQuery Functions and Operators 3.0, W3C Recommendation, 8 April 2014, <https://www.w3.org/TR/xpath-functions-30/>

XQuery 1.0 and XPath 2.0 Functions and Operators, W3C Recommendation, 14 December 2010, <https://www.w3.org/TR/2010/REC-xpath-functions-20101214/>

XQuery 3.0: An XML Query Language, 08 April 2014, <https://www.w3.org/TR/xquery-30/>

XQuery 3.1: An XML Query Language, 21 March 2017, <https://www.w3.org/TR/xquery-31/>

XSL Transformations (XSLT) Version 1.0, W3C Recommendation, 16 November 1999, <https://www.w3.org/TR/xslt-10/>

XSL Transformations (XSLT) Version 2.0, W3C Recommendation, 23 January 2007, <http://www.w3.org/TR/xslt20/>

XSL Transformations (XSLT) Version 3.0, W3C Recommendation, 8 June 2017, <https://www.w3.org/TR/xslt-30/>

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

abstract pattern

pattern (3.13) in a *Schematron schema* (3.7) that has been parameterized to enable reuse

3.2

abstract rule

collection of *assertions* (3.5) which can be included in other *rules* (3.18) but which does not fire itself

3.3

active pattern

pattern (3.13) belonging to the *active phase* (3.4)

3.4

active phase

phase (3.14) whose *patterns* (3.13) are used for validation

3.5

assertion

natural-language statement with associated *assertion test* (3.6) and ancillary attributes

3.6

assertion test

query returning a Boolean value

Note 1 to entry: An assertion test "succeeds" or "fails".

3.7

Schematron schema

document that satisfies all the requirements of this document

3.8

diagnostic

named natural-language statements providing information to end-users of validators concerning the expected and actual values together with repair hints

3.9

elaborated rule-context expression

single *rule-context expression* (3.20) which explicitly disallows items selected by lexically previous *rule contexts* (3.19) in the same *pattern* (3.13)

3.10

implementation

instantiation of a Schematron validator

3.11

name

mixture of characters with a restricted set of initial characters

Note 1 to entry: See Production 5 of XML 1.0 [\[3\]](#).

3.12

natural-language assertion

natural-language statement expressing some part of a *pattern* [\(3.13\)](#)

Note 1 to entry: A natural-language assertion is "met" or "unmet".

3.13

pattern

collection of *rules* [\(3.18\)](#) with an optional identifier and ancillary information

Note 1 to entry: Patterns of a schema do not have a defined order.

Note 2 to entry: Patterns are represented in schemas by elements *pattern* [\(5.4.12\)](#) and *group* [\(5.4.5\)](#).

3.14

phase

named, unordered collection of *patterns* [\(3.13\)](#)

Note 1 to entry: Patterns may belong to more than one phase.

Note 2 to entry: Three strings, #ALL, #ANY and #DEFAULT, are reserved with particular meanings.

3.15

progressive validation

validation of constraints in stages determined or grouped to some extent by the schema author rather than, for example, entirely determined by the document order

3.16

property

named data giving additional metadata on an *assertion* [\(3.5\)](#) or report

3.17

query language binding

named set, specified in a document, of the languages and conventions used for *assertion tests* [\(3.6\)](#), *rule-context expressions* [\(3.20\)](#) and so on, by a particular Schematron *implementation* [\(3.10\)](#)

3.18

rule

unordered collection of *assertions* [\(3.5\)](#) with a *rule-context expression* [\(3.20\)](#) and ancillary attributes

3.19

rule context

element or other information item used for *assertion tests* [\(3.6\)](#)

Note 1 to entry: A *rule* [\(3.18\)](#) is said to fire when an information item matches the rule context.

3.20

rule-context expression

query to specify *subjects* [\(3.21\)](#)

Note 1 to entry: A *rule context* [\(3.19\)](#) is said to match an information item when that information item has not been matched by any lexically-previous rule-context expressions in the same *pattern* [\(3.13\)](#) and the information item is one of the information items that the query would specify.

3.21**subject**

particular information item which corresponds to the object of interest of the *natural-language assertions* (3.12) and typically is matched by the context expression of a *rule* (3.18)

Note 1 to entry: Subjects can be more precisely defined in schemas by the `subject` attribute (5.5.19).

3.22**valid**

passing all *assertion tests* (3.6) in fired *rules* (3.18) of *active patterns* (3.3)

3.23**variable**

constant value, represented by a *name* (3.11), evaluated within the parent schema, *phase* (3.14), *pattern* (3.13) or *rule* (3.18) and scoped within the parent schema, phase, pattern or rule

Note 1 to entry: Variables are represented in schemas by elements `let` (5.4.7) and `param` (5.4.11).

4 Notation

This document uses XPath to identify information items, to the extent that items in the XPath data model can be derived from those defined by XML Infoset^[8], in Schematron schemas.

5 Syntax**5.1 Well-formedness**

A Schematron schema shall be a well-formed XML document, according to the version of XML (XML 1.0 or XML 1.1) used.

5.2 Namespace

All elements shown in the grammar for Schematron are qualified with the namespace URI [IRI^[9]]:

<http://purl.oclc.org/dsdl/schematron>

In this document, the prefix `sch` is taken as bound to the Schematron namespace URI for exposition purposes. The prefix `sch` is not reserved or required by this document. Any element can also have foreign attributes in addition to the attributes shown in the grammar. A foreign attribute is an attribute with a name whose namespace URI is neither the empty string nor the Schematron namespace URI. Any non-empty element may have foreign child elements in addition to the child elements shown in the grammar. A foreign element is an element with a name whose namespace URI is not the Schematron namespace URI. There are no constraints on the relative position of foreign child elements with respect to other child elements.

5.3 Whitespace

Any element can also have children strings that consist entirely of whitespace characters, where a whitespace character is one of U+0020, U+009, U+00D or U+00A. There are no constraints on the relative position of whitespace string children with respect to child elements.

Leading and trailing whitespace should be stripped from attributes defined by this document. Whitespace should be collapsed in elements defined by this document that allow text. Whitespace may be stripped from elements defined by this document that do not allow text.

5.4 Core elements

5.4.1 General

A Schematron schema shall follow the grammar given in [Annex A](#).

For information on the use of Schematron elements in external vocabularies, see [Annex F](#).

5.4.2 active element

The required `pattern` attribute is a reference to a pattern that is active in the current phase.

5.4.3 assert element

This element indicates an assertion made about the context nodes. The data content is a natural-language assertion. The required `test` attribute is an assertion test evaluated in the current context. If the test evaluates positive, the assertion succeeds. The optional `diagnostics` attribute is a reference to further diagnostic information. The optional `properties` attribute is a reference to additional arbitrary properties.

The natural-language assertion shall be a positive statement of a constraint.

The `icon`, `see` and `fpi` attributes ([5.5.9](#), [5.5.17](#) and [5.5.8](#)) allow rich interfaces and documentation.

The `flag` attribute ([5.5.6](#)) allows more detailed outcomes.

The `role` and `subject` attributes ([5.5.14](#) and [5.5.19](#)) allow explicit identification of some part of a pattern.

The optional `severity` attribute ([5.5.16](#)) expresses the relative importance of an assertion.

The natural-language assertion may contain information about actual values in addition to expected values and may contain diagnostic information. Users should note, however, that the `diagnostic` element is provided for such information to encourage clear statement of the natural-language assertion.

5.4.4 extends element

The `extends` element allows reference to the contents of other declarations. The `extends` element shall either have an `href` attribute or a `rule` attribute but not both.

Abstract rules are named lists of assertions without a context expression. An `extends` element with a `rule` attribute shall reference an abstract rule. The current rule uses all the assertions from the abstract rule it extends.

An `extends` element with an `href` attribute shall reference external declarations. An external declaration is a Schematron element of the same type as the parent element of the `extends` element: `schema`, `library` or `rule`.

NOTE There is no requirement for `rule` elements referenced using an `href` attribute to be abstract.

The `href` attribute is an IRI reference to an external well-formed XML document or to an element in an external well-formed XML document. If a document is referenced, its root element is the external declaration, otherwise the element referenced is the external declaration. The contents of the external declaration shall be inserted in place of the `extends` element.

In such a case, the relative position of elements in the post-inclusion document may be to that extent invalid against the schema for Schematron in [Annex A](#); however, other schema constraints such as containment shall still apply.

An implementation shall report any errors encountered during the processing of `extends`, for diagnostic purposes. Whether an implementation terminates in this event is implementation-defined.

5.4.5 group element

The `group` element has the same structure and semantics as element pattern (5.4.12), except that the if-then-else behaviour of evaluating rules within a pattern shall not apply (see 6.6).

5.4.6 include element

The required `href` attribute shall be an IRI reference to a well-formed XML document or to an element in a well-formed XML document.

The referenced element shall be inserted in place of the `include` element. The referenced element shall be a type which is allowed by the grammar for Schematron at the location of the `include` element.

An implementation shall report any errors encountered during the processing of `include`, for diagnostic purposes. Whether an implementation terminates in this event is implementation-defined.

5.4.7 let element

The `let` element declares a named variable. If the `let` element is the child of a `rule` element, the variable is calculated and scoped to the current rule and context. Otherwise, the variable is calculated with the context of the instance document root.

The required `name` attribute is the name of the variable. The `value` attribute is an expression evaluated in the current context. If no `value` attribute is specified, the value of the attribute is the element content of the `let` element.

The optional `as` attribute is the datatype of the variable. The query language binding shall determine whether the `as` attribute is supported and how it is processed.

It is an error to reference a variable that has not been defined in the current schema, phase, pattern or rule, if the query language binding allows this to be determined reliably. It is an error for a variable to be multiply defined in the current schema, phase, pattern and rule.

The variable is substituted into assertion tests and other expressions in the same rule before the test or expression is evaluated. The query language binding specifies which lexical conventions are used to detect references to variables.

5.4.8 library element

The `library` element is an allowed root element containing external declarations, including abstract rules (see 5.4.16) and patterns (see 5.4.12 and 5.4.5).

The contents of this element can be included in a schema by using `extends` (see 5.4.4).

5.4.9 name element

The `name` element provides the names of nodes from the instance document to allow clearer assertions and diagnostics. The optional `path` attribute is an expression evaluated in the current context that returns a string that is the name of a node. If the `path` attribute is omitted, the name of the context node is used.

5.4.10 ns element

The `ns` element specifies a namespace prefix and URI. The required `prefix` attribute is an XML name with no colon character. The required `uri` attribute is a namespace URI [IRI].

Namespace prefixes in context expressions, assertion tests and other query expressions shall be defined by this element. Namespace prefixes shall not use the namespace bindings in scope for element and attribute names.

NOTE 1 Namespace declarations as specified in XML 1.0 Namespaces^[10] (e.g. `xmlns:foo="http://www.example.com"`) do not apply to namespace prefixes in context expressions, assertion tests and other query expressions.

NOTE 2 Because the characters allowed as names can change in versions of XML subsequent to W3C XML 1.0^[3], the ISO/IEC 19757-2 ^[5] (RELAX NG Compact Syntax) schema for Schematron does not constrain the prefix to particular characters.

5.4.11 param element

The `param` element specifies a name-value pair providing parameters for an abstract pattern or a schema.

The required `name` attribute is an XML name with no colon. The `value` attribute is a fragment of a query.

The `value` attribute is required for parameters specified for schemas. The value of a parameter for a schema may be overridden at user option. This mechanism is implementation-defined.

The value of a parameter is specified for concrete patterns by the `value` attribute or if absent by the `value` attribute of a parameter whose `name` attribute matches that of a parameter for the abstract pattern the concrete pattern instantiates. It is an error if no parameter value is thereby specified for a concrete pattern. In a schema the names of pattern parameters and the number specified shall correspond between abstract and concrete patterns.

Parameter names shall be distinct within the scope of a pattern or schema.

5.4.12 pattern element

The `id` attribute provides a unique name for the pattern and is required for abstract patterns. The child `rule` elements of a pattern give constraints that are in some way related.

The optional `documents` attribute provides IRIs of the subordinate documents the rule contexts are relative to. The `documents` attribute is evaluated in the context of the original instance document root. Variables scoped to the pattern are not in scope when evaluating the `documents` attribute. The pattern is sought in each of the documents located at the IRI(s) which result from evaluating the expression and is not sought in the original instance document. Relative IRIs are resolved relative to the base IRI of the original instance document.

Conformant implementations should retrieve the subordinate documents in such a way that the results are deterministic. Where a document cannot be retrieved, the implementation shall report this for diagnostic purposes, and may do so by using element `svrl:error` (see [Annex D](#)). Whether the processor terminates in this event is implementation-defined.

The `icon`, `see` and `fpi` attributes ([5.5.9](#), [5.5.17](#) and [5.5.8](#)) allow rich interfaces and documentation.

When a `pattern` element has the attribute `abstract` with a value `true`, then the pattern defines an abstract pattern. An abstract pattern shall not have an `is-a` attribute and shall have an `id` attribute.

5.4.13 phase element

The `phase` element provides the means to declare named groups of patterns, for example to support progressive validation. Additional information on design requirements for progressive validation is provided in [Annex E](#).

The required `id` attribute is the name of the phase. The element specifies the phase to be used for validating documents, for example, by user command.

The optional `from` attribute (see [5.5.7](#)) can be used to restrict validation to a subset of a document.

The optional `when` attribute (see [5.5.22](#)) can be used to select the active phase based on the evaluation of its expression.

Three strings, `#ALL`, `#ANY` and `#DEFAULT`, have special meanings when specifying active phases. The string `#ALL` is reserved to denote that all patterns are active. The string `#ANY` denotes that the active phase is selected using the method given in [5.5.22](#) and if no active phase is selected, then all patterns are active. The

string `#DEFAULT` is to denote that the name given in the `defaultPhase` attribute on the `schema` element should be used. If no `defaultPhase` is specified, then all patterns are active.

The `icon`, `see` and `fpi` attributes ([5.5.9](#), [5.5.17](#) and [5.5.8](#)) allow rich interfaces and documentation.

The strings `#ALL`, `#ANY` and `#DEFAULT` shall not be used as phase names in a Schematron schema. They are for use when invoking or configuring schema validation, for example, as a command-line parameter.

5.4.14 `report` element

The `report` element specifies an assertion made about the context nodes. The data content is a natural-language assertion. The required `test` attribute is an assertion test evaluated in the current context. If the test evaluates positive, the report succeeds. The optional `diagnostics` attribute is a reference to further diagnostic information. The optional `properties` attribute is a reference to additional arbitrary properties.

The natural-language assertion shall be a positive statement of a found pattern or a negative statement of a constraint.

The `icon`, `see` and `fpi` attributes ([5.5.9](#), [5.5.17](#) and [5.5.8](#)) allow rich interfaces and documentation.

The `flag` attribute ([5.5.6](#)) allows more detailed outcomes.

The `role` and `subject` attributes ([5.5.14](#) and [5.5.19](#)) allow explicit identification of some part of a pattern.

The optional `severity` attribute ([5.5.16](#)) expresses the relative importance of an assertion.

The natural-language assertion may contain information about actual values in addition to expected values and may contain diagnostic information. Users should note, however, that the `diagnostic` element is provided for such information to encourage clear statement of the natural-language assertion.

5.4.15 `rule` element

The `rule` element provides a list of assertions tested within the context specified by the `context` attribute. The `context` attribute, required unless the `rule` element has the attribute `abstract` with a value `true`, specifies the rule-context expression.

The optional `visit-each` attribute (see [5.5.21](#)) refines the evaluation context.

The `icon`, `see` and `fpi` attributes ([5.5.9](#), [5.5.17](#) and [5.5.8](#)) allow rich interfaces and documentation.

The `flag` attribute ([5.5.6](#)) allows more detailed outcomes.

The `role` and `subject` attributes ([5.5.14](#) and [5.5.19](#)) allow explicit identification of some part of a pattern as part of the validation outcome.

When the `rule` element has the attribute `abstract` with a value `true`, then the rule is an abstract rule. An abstract rule shall not have a `context` attribute. An abstract rule is a list of assertions that will be invoked by other rules, using the `extends` element. Abstract rules provide a mechanism for reducing schema size.

It is not an error if a rule never fires in a document. In order to test that a document always has some context, a new pattern should be created from the context of the document, with an assertion requiring the element or attribute.

5.4.16 `rules` element

The `rules` element is a container for abstract rules, allowing their use across patterns.

This element may be used as a top-level element in a schema and can also be used in a `library` ([5.4.8](#)).

5.4.17 `schema` element

The `schema` element is the top-level element of a Schematron schema.

The optional `schemaVersion` attribute gives the version of the schema. Its allowed values are not defined by this document and its use is implementation-dependent.

The optional `queryBinding` attribute provides the short name of the query language binding in use. The `defaultPhase` attribute may be used to indicate the phase to use in the absence of explicit user-supplied information.

The `icon`, `see` and `fpi` attributes ([5.5.9](#), [5.5.17](#) and [5.5.8](#)) allow rich interfaces and documentation.

5.4.18 value-of element

The `value-of` element finds or calculates values from the instance document to allow clearer assertions and diagnostics. The required `select` attribute is an expression evaluated in the current context that returns a string.

Variable references in the `select` attribute are resolved in the scope of the current schema, phase, pattern and rule.

5.5 Ancillary elements and attributes

5.5.1 diagnostic element

The `diagnostic` element provides a natural-language message giving more specific details concerning a failed assertion, such as found versus expected values and repair hints.

The variables in scope when evaluating assertion tests shall also be in scope when evaluating diagnostic elements associated with the assertion.

NOTE 1 Diagnostics in multiple languages can be supported by using a different `diagnostic` element for each language, with the appropriate `xml:lang` language attribute, and referencing all the unique identifiers of the `diagnostic` elements in the `diagnostics` attribute of the assertion. [Annex G](#) gives a simple example of a multi-lingual schema.

NOTE 2 Typical values for the `role` attribute on a `diagnostic` element can be `warning`, `caution` or `note`.

5.5.2 diagnostics attribute

The `diagnostics` attribute associates an `assert` or `report` element with further diagnostic information (see [5.5.1](#)).

5.5.3 diagnostics element

The `diagnostics` element provides a section containing individual `diagnostic` elements.

5.5.4 dir element

The `dir` element provides a section of natural-language text with a direction specified by the `value` attribute. The value `ltr` indicates left-to-right text; the value `rtl` indicates right-to-left text.

5.5.5 emph element

The `emph` element encloses a portion of text that should be rendered with some emphasis.

5.5.6 flag attribute

The `flag` attribute acts as a Boolean value with the initial value `false`. A flag is implicitly declared by an assertion or rule having a `flag` attribute with that name. The value of a flag becomes `true` when an assertion with that flag fails or a rule with that flag fires.

The purpose of flags is to convey state or severity information to a subsequent process.