INTERNATIONAL STANDARD

ISO/IEC 19757-5

First edition
2011-10-01

Information technology — Document Schema Definition Languages (DSDL) —

Part 5: Extensible Datatypes

Technologies de l'information — Langages de définition de schéma de documents (DSDL) —
Partie 5: Types de données extensibles

Reference number
ISO/IEC 19757-5:2011(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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

ISO/IEC 19757-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*.

ISO/IEC 19757 consists of the following parts, under the general title *Information technology — Document Schema Definition Languages (DSDL)*:

— *Part 1: Overview*

— *Part 2: Regular-grammar-based validation — RELAX NG*

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

— *Part 4: Namespace-based Validation Dispatching Language (NVDL)*

— *Part 5: Extensible Datatypes*

— *Part 7: Character Repertoire Description Language (CREPDL)*

— *Part 8: Document Semantics Renaming Language (DSRL)*

— *Part 9: Namespace and datatype declaration in Document Type Definitions (DTDs)*

— *Part 11: Schema Association*

Introduction

This part of ISO/IEC 19757 specifies a powerful, XML-based language which enables users to create and extend their own libraries of datatypes using straightfoward declarative XML constructs. Such libraries are well-suited to being used in pipelining validation processes in conjunction with other XML schema languages.

Unlike W3C Schema[1], ISO/IEC 19757-2:2008 (RELAX NG) does not itself provide a declarative mechanism for users to define their own datatypes. If they are not satisfied with the two built-in types of `string` and `token`, RELAX NG users have had either to use a pre-written library bundled with their validator, or to program a datatype library using that validator's API. Such programmed datatype libraries are hard to construct for non-programmer users, and built-in datatype libraries are often insufficient for users' needs.

Information technology — Document Schema Definition Languages (DSDL) —

Part 5: Extensible Datatypes

1 Scope

This part of ISO/IEC 19757 specifies an XML language that allows users to create and extend datatype libraries for their own purposes. The datatype definitions in these libraries can be used by XML validators and other tools to validate content and make comparisons between values.

2 Normative references

The following referenced documents are indispensable for the application 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.

IETF RFC 3023, *XML Media Types*, Internet Standards Track Specification, January 2001, http://www.ietf.org/rfc/rfc3023.txt

IETF RFC 3987, *Internationalized Resource Identifiers (IRIs)*, Internet Standards Track Specification, January 2005, http://www.ietf.org/rfc/rfc3987.txt

ISO/IEC 19757-2:2008, *Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG*

W3C XML, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation, 16 August 2006, edited in place 29 September 2006, http://www.w3.org/TR/2006/REC-xml-20060816

W3C XML Names, *Namespaces in XML 1.0 (Third Edition)*, W3C Recommendation, 8 December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

W3C XPath 2.0, *XML Path Language (XPath) 2.0*, W3C Recommendation, 23 January 2007, http://www.w3.org/TR/2007/REC-xpath20-20070123/

W3C XPath 2.0 Functions, *XQuery 1.0 and XPath 2.0 Functions and Operators*, W3C Recommendation, 23 January 2007, http://www.w3.org/TR/2007/REC-xpath-functions-20070123/

W3C XSLT 2.0, *XSL Transformations (XSLT) Version 2.0*, W3C Recommendation, 23 January 2007, http://www.w3.org/TR/2007/REC-xslt20-20070123/

W3C XLink 1.0, *XML Linking Language (XLink) Version 1.0*, W3C Recommendation, 27 June 2001, http://www.w3.org/TR/2001/REC-xlink-20010627/

3 Terms and definitions

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

3.1
candidate value
some character data in an XML document that is to have its datatype tested

3.2
datatype
named property of sequences of character data that validate against datatype definitions as described by this part of ISO/IEC 19757

3.3
datatype definition
formal specification of constraints upon XML character data for the datatype being defined

3.4
datatype library
collection of datatype definitions that share the same XML Namespace

3.5
Extensible Datatypes document
XML document which is valid to the normative schema presented in this part of ISO/IEC 19757, and which conforms to its provisions

3.6
forwards-compatible mode
operating mode in which an implementation ignores language constructs which are labelled as having a version later than that understood by the implementation, unless they are explicitly labelled as requiring processing

3.7
implementation
Extensible Datatypes implementation that conforms to this part of ISO/IEC 19757

3.8
extended implementation
implementation that conforms to this part of ISO/IEC 19757, and which provides additional functionality provided by the extension mechanisms of Extensible Datatypes

4 Extensible Datatypes schema overview

The schema for Extensible Datatypes is interspersed as fragments within the narrative text of this part of ISO/IEC 19757 and appears rendered against a grey background. The schema language used is the compact syntax of RELAX NG, as defined by Annex C of ISO/IEC 29500:2008.

Concatenating the schema fragments in this part of ISO/IEC 19757 gives a RELAX NG schema that normatively defines the grammar of Extensible Datatypes. The consolidated schema is shown in Annex A.

NOTE 1 Throughout, as per ISO/IEC 19757-2:2008, RELAX NG compact syntax keywords used as identifiers in the schema are prefixed with the "\" character.

NOTE 2 The null Namespace is bound to the prefix `"local"` so that it can be referenced later in the schema.

```
default namespace dt =
          "http://purl.oclc.org/dsdl/extensible-datatypes"
namespace local = ""
```

Datatype libraries are defined in ISO/IEC 19757-2:2008 as being identified by an IRI, with each datatype within a given datatype library being identified by a NCName. An Extensible Datatypes document presents one or more such datatype libraries to implementations. Each datatype definition has a qualified name; the Namespace IRI identifies the datatype library to which the datatype belongs, and the local part identifies the name of the datatype within that datatype library.

5 Common constructs

5.1 Common types

5.1.1 XPath expressions

W3C XPath 2.0 expressions are used to bind values to variables or properties and to express tests in conditions. Conforming implementations of Extensible Datatypes shall additionally implement the following functions from W3C XSLT 2.0:

— `document` (W3C XSLT 2.0, section 16.1)

— `format-number` (W3C XSLT 2.0, section 16.4)

— `function-available` (W3C XSLT 2.0, section 18.1.1)

These functions shall be callable within implementations using unqualified function names.

NOTE The `function-available` function can be used to test for the availability of XPath extension functions for the purposes of enhanced validation in extended implementations.


```
XPath = text
```


The context node for evaluating XPath expressions in Extensible Datatypes is a text node that is the only child of a root node, and whose value is the whitespace-normalized candidate value. The context position and context size are both 1. The set of variable bindings are the in-scope variables, as defined at 9.4.1. The set of namespace declarations that are in-scope for the expression are those that are in-scope for the element on which the XPath is given.

5.1.2 Boolean values

Where a boolean value is to be specified, the literal strings "`true`" and "`false`" are used.

```
boolean = "true" | "false"
```

5.1.3 Regular expressions

Regular expressions are defined in W3C XPath 2.0.

```
regular-expression = text
```

5.1.4 Arbitrary content

Extensible Datatypes document are governed by an open schema which, for purposes of extensibility, allows arbitrary content to occur at certain points. Such content can be any XML content other than elements or attributes associated with the Extensible Datatypes XML Namespace.

```
anything =
 mixed {
   element * - dt:* {
     attribute * - dt:* { text }*,
     anything
   }*
 }
```

5.2 Common attributes

5.2.1 `version` attribute

The value of the `version` attribute specifies the version of Extensible Datatypes being used within the element on which it occurs. The version described by this International Standard is "1.0"

5.2.2 `ns` attribute

The value of the `ns` attribute specifies the Namespace IRI of those datatypes defined within that element whose `name` attribute does not include a prefix, thus determining the datatype library to which these datatypes belong. This value shall be an IRI as defined by IETF RFC 3987.

```
ns = attribute ns { text }
```

5.2.3 `name` attribute

The `name` attribute specifies the name of a datatype, parameter, variable or property. The value of a `name` attribute is a qualified name. If no prefix is specified the Namespace IRI associated with the name depends on the element on which the `name` attribute occurs. If the `name` attribute occurs on a `datatype` element, the Namespace IRI is that given in the `ns` attribute of the `datatype` element or its nearest ancestor element that has a `ns` attribute, if there is one, or no Namespace IRI if there is not. Otherwise, the unprefixed name has no Namespace IRI.

NOTE Names conform to the `text` pattern defined by ISO/IEC 29500-2:2008 and so are less constrained that names in some other schema specifications.

```
name = attribute name { text }
```

5.2.4 Extension attributes

Extension attributes are attributes in any non-null Namespace other than the Extensible Datatypes namespace. They can occur on any Extensible Datatypes element. The presence of such attributes shall not change the behaviour of the Extensible Datatypes elements defined in this part of ISO/IEC 19757: an implementation shall return the same result whether extension attributes are processed or not.

```
extension-attribute = attribute * - (local:* | dt:*) { text }
```

5.3 Extension elements

Extension elements are elements in any Namespace other than the Extensible Datatypes Namespace. They may be processed by extended implementations. There are three classes of extension element:

— top-level extension elements, which occur as children of the document element or `div` elements

— definition extension elements, which occur as children of `datatype` elements

— binding extension elements, which occur wherever a value can be bound (for example, to a variable)

```
extension-element =
  element * - dt:* {
    must-implement?,
    attribute * - ( dt:* | must-implement) { text }*,
    anything
  }
```

5.4 Versioning and compatibility

An Extensible Datatypes element is processed in forwards-compatible mode if it, or its nearest ancestor that has a `version` attribute, has a `version` attribute with a value greater than "1.0". When an element in the Extensible Datatypes namespace that is not described by this part of ISO/IEC 19757 is processed in forwards-compatible mode it, its attributes and its descendants shall be ignored unless it has a `must-implement` attribute with the value `true`, in which case an implementation shall halt and emit an error message.

```
must-implement = attribute must-implement { boolean }
```

6 Simplification

Before it is applied for validation, an Extensible Datatypes document is simplified into a single logical unit by processing any `include` elements and resolving any multiple occurrences of same-named data types into a single definition.

6.1 Include elements

`include` elements reference other Extensible Datatypes datatype libraries. They import datatypes from these libraries or redefine them using definitions in the host document.


```
\include =
      element include {
      ns?,
      attribute href { text },
      extension-attribute*,
      top-level-element*
      }
```

The `ns` attribute on `include` is used to override the namespace of imported datatypes as defined using the `ns` attribute in the document being included.

The `href` attribute specifies an IRI reference. This IRI reference is first transformed by escaping disallowed characters as specified in Section 5.4 of W3C XLink 1.0. If it is not absolute, the IRI reference is resolved into an absolute form as described in section 5 of IETF RFC 3987 using the base IRI of the `include` element.

The value of the `href` attribute is thus used to create a `datatypes` element, as follows. The IRI reference consists of the IRI itself and an optional fragment identifier. The resource identified by the IRI is retrieved. The result is a MIME entity: a sequence of octets labeled with a MIME media type. The media type determines how an element is constructed from the MIME entity and optional fragment identifier. When the media type is `application/xml` or `text/xml`, the MIME entity shall be parsed as an XML document in accordance with the applicable RFC (at the time of writing [RFC 3023]) and an element, which shall be a `datatypes` element in the Extensible Datatypes namespace, constructed from the result of the parse. In particular, the `charset` parameter shall be handled as specified by the RFC. This specification does not define the handling of media types other than `application/xml` and `text/xml`. The `href` attribute shall not include a fragment identifier unless the registration of the media type of the resource identified by the attribute defines the interpretation of fragment identifiers for that media type.

NOTE [RFC 3023] does not define the interpretation of fragment identifiers for `application/xml` or `text/xml`.