XML Schema - Data Types Quick Reference

ver 9/01

http://www.xml.dvint.com

1 Namespaces

- http://www.w3.org/2001/XMLSchema
- http://www.w3.org/2001/XMLSchema-datatypes

2 Logic Types

boolean binary-valued logic legal literals {true, false, 1, 0}

3 Binary Data Types

base64Binary Base64-encoded arbitrary binary data.

hexBinary Arbitrary hex-encoded binary data. Example, "0FB7" is a hex encoding for 16-bit int 4023 (binary 111110110111).

4 Text types

anyURI A Uniform Resource Identifier Reference (URI). Can be absolute or relative, and may have an optional fragment identifier

language natural language identifiers [RFC 1766] Example: en, fr

normalizedString White space normalized strings

string Character strings in XML

token Tokenized strings.

5 Number Types

byte 127 to-128. Sign is omitted, "+" assumed.
Example: -1, 0, 126, +100.

decimal Arbitrary precision decimal numbers. Sign omitted, "+" is assumed. Leading and trailing zeroes are optional. If the fractional part is zero, the period and following zero(es) can be omitted.

double Double-precision 64-bit floating point type - legal literals {0, -0, INF, -INF and NaN} Example, -1E4, 12.78e-2, 12 and INF

float 32-bit floating point type - legal literals {0, -0, INF, -INF and NaN} Example, -1E4, 1267.43233E12, 12.78e-2, 12 and INF

int 2147483647 to -2147483648. an optional sign followed by a finite-length sequence of decimal digits (#x30-#x39). If the sign is omitted, "+" is assumed. Example: -1, 0, 126789675, +100000.

integer Integer or whole numbers - Sign omitted, "+" is assumed. Example: -1, 0, 12678967543233, +100000.

long 9223372036854775807 to -9223372036854775808. Sign omitted, "+" assumed.
Example: -1, 0, 12678967543233, +100000.

negativeInteger Infinite set {...,-2,-1}.
Example: -1, -12678967543233, -100000.

nonNegativeInteger Infinite set {0, 1, 2, ...}. Sign omitted, "+" assumed, leading zeroes are prohibited.
Example: 1, 0, 12678967543233, +100000.

nonPositiveInteger Infinite set {...,-2,-1,0}. Example: -1, 0, -126733, -100000.

positiveInteger Infinite set {1, 2, ...}. Optional "+" sign, leading zeroes are prohibited. Example: 1, 12678967543233, +100000.

short 32767 to -32768. Sign omitted, "+" assumed.
Example: -1, 0, 12678, +10000.

unsignedByte 0 to 255. a finite-length leading zeroes prohibited.
Example: 0, 126, 100.

unsignedInt 0 to 4294967295 leading zeroes are prohibited. Example: 0, 1267896754, 100000.

unsignedLong 0 to 18446744073709551615.
Example: 0, 12678967543233, 100000.

unsignedShort 0 to 65535 leading zeroes are prohibited.
Example: 0, 12678, 10000.

6 Date Time Types

date Calendar date. Example, May the 31st, 1999 is: 1999-05-31.

dateTime Specific instant of time. ISO 8601 extended format CCYY-MM-DDThh:mm:ss. Example, to indicate 1:20 pm on May the 31st, 1999 for Eastern Standard Time which is 5 hours behind Coordinated Universal Time (UTC): 1999-05-31T13:20:00-05:00.

duration A duration of time. ISO 8601 extended format
PnYn MnDTnH nMn S. Example, to indicate duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes: P1Y2M3DT10H30M. One could also indicate a duration of minus 120 days as: -P120D.

gDay Gregorian day, a day such as the 5th of the month.

gMonth Gregorian month. Example: May is 05.

gMonthDay Gregorian specific day in a month.
Example: Feb 5 is 02-05.

gYear Gregorian calendar year. Example, year 1999, write: 1999.

gYearMonth Specific gregorian month and year.
Example, May 1999, write: 1999-05.

time An instant of time that recurs every day. Example, 1:20 pm for Eastern Standard Time which is 5 hours behind Coordinated Universal Time (UTC), write: 13:20:00-05:00.

7 XML Types

Name XML Names

NCName XML "non-colonized" Names.

NOTATION NOTATION type

QName XML qualified names

Following attribute types should only be used in attribute declaration for compatiblity reasons:

ENTITIES ENTITIES attribute type

ENTITY ENTITY attribute type

ID ID attribute type

IDREF IDREF attribute type

IDREFS IDREFS attribute type

NMTOKEN NMTOKEN attribute type

NMTOKENS NMTOKENS attribute type

8 Simple Data Type Declaration

<simpleType id = ID
final = (#all | (list | union | restriction))
name = NCName>
Content: (annotation ?, (restriction | list | union)) </simpleType>

<restriction id = ID
base = QName>
Content: (annotation ?, (simpleType ?, (minExclusive | minInclusive | maxExclusive | maxInclusive | totalDigits | fractionDigits | length | minLength | maxLength | enumeration | whiteSpace | pattern)*)) </restriction>

<list id = ID
itemType = QName>
Content: (annotation ?, (simpleType ?)) </list>
<union id = ID
memberTypes = List of QName>
Content: (annotation ?, (simpleType *)) </union>

xmlhelp@dvint.com
http://www.xml.dvint.com

Constraining Facets

<length id = ID
fixed = boolean : false
value = nonNegativeInteger >
Content: (annotation?) </length>

<minLength id = ID
fixed = boolean : false
value = nonNegativeInteger >
Content: (annotation?) </minLength>

<maxLength id = ID
fixed = boolean : false
value = nonNegativeInteger >
Content: (annotation?) </maxLength>

<pattern id = ID
value = anySimpleType
Content: (annotation?) </pattern>

<enumeration id = ID
value = anySimpleType >
Content: (annotation?)
</enumeration>

<whiteSpace id = ID
fixed = boolean : false
value = (collapse | preserve | replace)>
Content: (annotation?)
</whiteSpace>

<maxInclusive id = ID
fixed = boolean : false
value = anySimpleType>
Content: (annotation?)
</maxInclusive>

<maxExclusive id = ID
fixed = boolean : false
value = anySimpleType>
Content: (annotation?)
</maxExclusive>

<minExclusive id = ID
fixed = boolean : false
value = anySimpleType>
Content: (annotation?)
</minExclusive>

<minInclusive id = ID
fixed = boolean : false
value = anySimpleType>
Content: (annotation?)
</minInclusive>

<totalDigits id = ID
fixed = boolean : false
value = positiveInteger >
Content: (annotation?)
</totalDigits>

<fractionDigits id = ID
fixed = boolean : false
value = nonNegativeInteger >
Content: (annotation?)
</fractionDigits>

Data Type	length	minLength	maxLength	pattern	enumeration	whiteSpace	maxInclusive	maxEclusive	minExclusive	minInclusive	totalDigits	fractionDigits
anyURI	✓	✓	✓	✓	✓	✓						
base64Binary	✓	✓	✓	✓	✓	✓						
boolean				✓		✓						
byte			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
date				✓	✓	✓	✓	✓	✓	✓		
dateTime				✓	✓	✓	✓	✓	✓	✓		

Data Type	length	minLength	maxLength	pattern	enumeration	whiteSpace	maxInclusive	maxEclusive	minExclusive	minInclusive	totalDigits	fractionDigits
decimal				✓	✓	✓	✓	✓	✓	✓	✓	✓
double				✓	✓	✓	✓	✓	✓	✓		
duration				✓	✓	✓	✓	✓	✓	✓		
ENTITIES	✓	✓	✓		✓	✓						
ENTITY	✓	✓	✓	✓	✓	✓						
float				✓	✓	✓	✓	✓	✓	✓		
gDay				✓	✓	✓	✓	✓	✓	✓		
gMonth				✓	✓	✓	✓	✓	✓	✓		
gMonthDay				✓	✓	✓	✓	✓	✓	✓		
gYear				✓	✓	✓	✓	✓	✓	✓		
gYearMonth				✓	✓	✓	✓	✓	✓	✓		
hexBinary	✓	✓	✓	✓	✓	✓						
ID	✓	✓	✓	✓	✓	✓						
IDREF	✓	✓	✓	✓	✓	✓						
IDREFS	✓	✓	✓		✓	✓						
int			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
integer			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
language	✓	✓	✓	✓	✓	✓						
long			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Name	✓	✓	✓	✓	✓	✓						
NCName	✓	✓	✓	✓	✓	✓						
negativeInteger			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NMTOKEN	✓	✓	✓	✓	✓	✓						
NMTOKENS	✓	✓		✓	✓	✓						
nonNegativeInteger			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
nonPositiveInteger			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
normalizedString	✓	✓	✓	✓	✓	✓						
NOTATION	✓	✓	✓	✓	✓	✓						
positiveInteger			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QName	✓	✓	✓	✓	✓	✓						
short			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
string	✓	✓	✓	✓	✓	✓						
time				✓	✓	✓	✓	✓	✓	✓		
token	✓	✓	✓	✓	✓	✓						
unsignedByte			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
unsignedInt			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
unsignedLong			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
unsignedShort			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

9 Regular Expressions for Patterns

Special Characters needing to be escaped with a '\'

\ | . - ^ ? * + { } ( ) [ ]

Special Character Sequences

\n newline

\r return

\t tab

. (dot) all characters except newline and return

\s space characters (space, tab, newline, return)

\S non-Space characters

\i initial XML name characters (letter _ ;)

\I not initial XML name characters

\c XML NameChar characters

\C not XML NameChar characters

\d decimal digits

\D not decimal digits

\w XML Letter or Digit characters

\W not XML Letter or Digit characters

\p{IsBasicLatin} block escape identifying ASCII characters, similar IsGreek, IsHebrew, IsThai for these ranges of Unicode blocks

\p{L} all Letters

\p{M} all Marks

\p{N} all Numbers

\p{P} all Punctuation

\p{Z} all Separators

\p{S} all Symbols

\p{C} all Others. Additional modifying values like Lu = uppercase, Ll = lowercase, Nd = decimal digit, Sm = math symbols, Sc = currency

\P{} not the block or category, \P{IsGreek} = not Greek block

Character References

N or c for hex or decimal XML character references

Repetition Operators

***** = 0 or more, **?** 0 or 1, **+** 1 or more

Interval Operators

{x,y} range x to y, {x,} at least x, {x} exactly x, i.e. {4,8} 4 to 8

Range Expressions

[a-zA-Z] = character a to z upper and lower case
[0-9] = digits 0 to 9