XML Schema - Data Types Quick Reference

ver 1/03

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

1 Namespaces §3.1 pt2

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

2 Logic Types

boolean	atomic	binary-valued logic legal literals {true, false, 1, 0}	§3.2.1.2 pt2

3 Binary Data Types

base64Binary	atomic	Base64-encoded arbitrary binary data.	§3.2.16 pt2
hexBinary	atomic	Arbitrary hex-encoded binary data. Example, “0FB7” is a hex encoding for 16-bit int 4023 (binary 111110110111).	§3.2.15 pt2

4 Text types

anyURI	atomic	A Uniform Resource Identifier Reference (URI). Can be absolute or relative, and may have an optional fragment identifier.	§3.2.17 pt2
language	derived token	natural language identifiers [RFC 1766] Example: en, fr.	§3.3.3 pt2
normalizedString	derived string	White space normalized strings	§3.3.1 pt2
string	atomic	Character strings in XML	§3.2.1 pt2
token	derived normalized-String	Tokenized strings.	§3.3.2 pt2

5 Number Types

byte	derived short	127 to-128. Sign is omitted, “+” assumed. Example: -1, 0, 126, +100.	§3.3.19 pt2
decimal	atomic	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.	§3.2.3 pt2
double	atomic	Double-precision 64-bit floating point type - legal literals {0, -0, INF, -INF and NaN} Example, -1E4, 12.78e-2, 12 and INF	§3.2.5 pt2
float	atomic	32-bit floating point type - legal literals {0, -0, INF, -INF and NaN} Example, -1E4, 1267.43233E12, 12.78e-2, 12 and INF	§3.2.4 pt2
int	derived long	2147483647 to -2147483648. Sign omitted, “+” is assumed. Example: -1, 0, 126789675, +100000.	§3.3.17 pt2
integer	derived decimal	Integer or whole numbers - Sign omitted, “+” is assumed. Example: -1, 0, 12678967543233, +100000.	§3.3.13 pt2
long	derived integer	9223372036854775807 to -9223372036854775808. Sign omitted, “+” assumed. Example: -1, 0, 12678967543233, +100000.	§3.3.16 pt2
negativeInteger	derived nonPositive	Infinite set {...,-2,-1}. Example: -1, -12678967543233, -100000.	§3.3.15 pt2
nonNegativeInteger	derived integer	Infinite set {0, 1, 2,...}. Sign omitted, “+” assumed. Example: 1, 0, 12678967543233, +100000.	§3.3.20 pt2
nonPositiveInteger	derived integer	Infinite set {...,-2,-1,0}. Example: -1, 0, -126733, -100000.	§3.3.14 pt2
positiveInteger	derived nonNegativeInteger	Infinite set {1, 2,...}. Optional “+” sign,. Example: 1, 12678967543233, +100000.	§3.3.25 pt2
short	derived int	32767 to -32768. Sign omitted, “+” assumed. Example: -1, 0, 12678, +10000.	§3.3.18 pt2
unsignedByte	derived unsignedShort	0 to 255. a finite-length Example: 0, 126, 100.	§3.3.24 pt2
unsignedInt	derived unsignedLong	0 to 4294967295 Example: 0, 1267896754, 100000.	§3.3.22 pt2
unsignedLong	derived nonNegative	0 to 18446744073709551615. Example: 0, 12678967543233, 100000.	§3.3.21 pt2
unsignedShort	derived unsignedInt	0 to 65535. Example: 0, 12678, 10000.	§3.3.23 pt2

6 Date Time Types

date	atomic	Calendar date.Format CCYY-MM-DD. Example, May the 31st, 1999 is: 1999-05-31.	§3.2.9 pt2
dateTime	atomic	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.	§3.2.7 pt2
duration	atomic	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.	§3.2.6 pt2
gDay	atomic	Gregorian day. Example a day such as the 5th of the month is --05.	§3.2.13 pt2
gMonth	atomic	Gregorian month. Example: May is --05--	§3.2.14 pt2
gMonthDay	atomic	Gregorian specific day in a month. Example: Feb 5 is --02-05.	§3.2.12 pt2
gYear	atomic	Gregorian calendar year. Example, year 1999, write: 1999.	§3.2.11 pt2
gYearMonth	atomic	Specific gregorian month and year. Example, May 1999, write: 1999-05.	§3.2.10 pt2
time	atomic	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.	§3.2.8 pt2

7 XML Types

Name	derived token	XML Names	§3.3.6 pt2
NCName	derived Name	XML “non-colonized” Names.	§3.3.7 pt2
NOTATION	atomic	NOTATION type	§3.2.19 pt2
QName	atomic	XML qualified names	§3.2.18 pt2

Following types should only be used in attribute declaration for XML compatibility:

ENTITY	derived NCName	ENTITY attribute type	§3.3.11 pt2
ENTITIES	derived ENTITY	ENTITIES attribute type	§3.3.12 pt2
ID	derived NCNAME	ID attribute type	§3.3.8 pt2
IDREF	derived NCName	IDREF attribute type	§3.3.9 pt2
IDREFS	derived IDREF	IDREFS attribute type	§3.3.10 pt2
NMTOKEN	derived token	NMTOKEN attribute type	§3.3.4 pt2
NMTOKENS	derived NMTOKENS	NMTOKENS attribute type	§3.3.5 pt2

8 Built-in Types

anyType	ur-type definition	Built-in Complex type definition of Ur-Type.	§3.4.7 pt1
anySimpleType	ur-type definition	Built-in Simple type definition of Ur-Type.	§4.1.6 pt2

9 Simple Data Type Declaration §4.1.2 pt2

Note: All schema components allow attributes from non-schema namespaces.

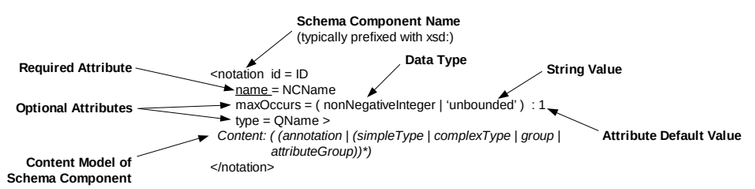
<simpleType id = ID
final = (‘#all’| (‘list’| ‘union’| ‘restriction’))
name = NCName>
Content: (annotation ?, (restriction | list | union)) </simpleType>

<list id = ID
itemType = QName>
Content: (annotation ?, (simpleType ?)) </list>

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

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

.



10 Constraining Facets §4.3 pt2

<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>

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

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

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

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

11 Simple Data Types and Constraining Facets §4.1.5 pt2, Appendix B pt0

Simple Data Type	length	minLength	maxLength	pattern	enumeration	whiteSpace	maxInclusive	maxExclusive	minExclusive	minInclusive	totalDigits	fractionDigits
anyURI	u	u	u	u	u	u						
base64Binary	u	u	u	u	u	u						
boolean				u		u						
byte - 127 to-128				u	u	u	u	u	u	u	u	u
date - CCYY-MM-DD				u	u	u	u	u	u	u		
dateTime - CCYY-MM-DDThh:mm:ss				u	u	u	u	u	u	u		
decimal - Arbitrary precision decimal numbers				u	u	u	u	u	u	u	u	u
double - Double-precision 64-bit floating point				u	u	u	u	u	u	u		
duration - PnYn MnDTnH nMn S				u	u	u	u	u	u	u		
ENTITIES	u	u	u		u	u						
ENTITY	u	u	u	u	u	u						
float - 32-bit floating point type				u	u	u	u	u	u	u		
gDay				u	u	u	u	u	u	u		
gMonth				u	u	u	u	u	u	u		
gMonthDay				u	u	u	u	u	u	u		
gYear				u	u	u	u	u	u	u		
gYearMonth				u	u	u	u	u	u	u		
hexBinary	u	u	u	u	u	u						
ID	u	u	u	u	u	u						
IDREF	u	u	u	u	u	u						
IDREFS	u	u	u		u	u						
int - 2147483647 to -2147483648.				u	u	u	u	u	u	u	u	u
integer				u	u	u	u	u	u	u	u	u
language - RFC 1766] Example: en, fr	u	u	u	u	u	u						
list	u	u	u	u	u	u						
long - 9223372036854775807 to -9223372036854775808				u	u	u	u	u	u	u	u	u
Name	u	u	u	u	u	u						
NCName	u	u	u	u	u	u						
negativeInteger				u	u	u	u	u	u	u	u	u
NMTOKEN	u	u	u	u	u	u						
NMTOKENS	u	u	u		u	u						
nonNegativeInteger				u	u	u	u	u	u	u	u	u
nonPositiveInteger				u	u	u	u	u	u	u	u	u
normalizedString	u	u	u	u	u	u						
NOTATION	u	u	u	u	u	u						
positiveInteger				u	u	u	u	u	u	u	u	u
QName	u	u	u	u	u	u						
short - 32767 to -32768				u	u	u	u	u	u	u	u	u
string	u	u	u	u	u	u						
time - hh:mm:ss				u	u	u	u	u	u	u		
token	u	u	u	u	u	u						
union				u	u							
unsignedByte - 0 to 255				u	u	u	u	u	u	u	u	u
unsignedInt - 0 to 4294967295				u	u	u	u	u	u	u	u	u
unsignedLong - 0 to 18446744073709551615				u	u	u	u	u	u	u	u	u
unsignedShort - 0 to 65535				u	u	u	u	u	u	u	u	u

12 Regular Expressions for Pattern Facet §4.3.4 pt2 §Appendix D pt0, §Appendix F pt2

Special Characters needing to be escaped with a '\'

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

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

Character Range Expressions

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

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

Pattern Examples

Expression	Match(es)
Chapter \d	Chapter 0, Chapter 1, Chapter 2....
Chapter\s\w	Chapter followed by a single whitespace character (space, tab, newline, etc.), followed by a word character (XML 1.0 Letter or Digit)
Espanñola	Española
\p{Lu}	any uppercase character, the value of \p{} (e.g. "Lu") is defined by Unicode
a*x	x, ax, aax, aaax....
a?x	ax, x
a+x	ax, aax, aaax....
(a|b)+x	ax, bx, aax, abx, bax, bbx, aaax, aabx, abax, abbx, baax, babx, bbax, bbbx, aaaax....
[^0-9]x	any non-digit character followed by the character x
\Dx	any non-digit character followed by the character x
.x	any character followed by the character x
.*abc.*	1x2abc, abc1x2, z3456abchooray....
ab{2}x	abbx
ab{2,4}x	abbx, abbbx, abbbbx
ab{2,}x	abbx, abbbx, abbbbx....
(ab){2}x	ababx

ver 1/03

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