

XML Schema - Structures Quick Reference	
ver 1/03 ©2002, 2003 D Vint Productions xmlhelp@dvint.com http://www.xml.dvint.com	
Note: All schema components allow attributes from non-schema namespaces.	
1Namespaces - http://www.w3.org/2001/XMLSchema - http://www.w3.org/2001/XMLSchema-instance	\$2.6 pt1
2Schema Declaration <schema id = ID attributeFormDefault = ('qualified' 'unqualified') : 'unqualified' blockDefault = ('#all' List of ('extension' 'restriction' 'substitution')) : '' elementFormDefault = ('qualified' 'unqualified') : 'unqualified' finalDefault = ('#all' List of ('extension' 'restriction')) : '' targetNamespace = anyURI version = token xml:lang = language > Content: ((include import redefine annotation)*, (((simpleType complexType group attributeGroup) element attribute notation), annotation*)*) </schema>	\$3.15.2 pt1
3Schema Management <include id = ID schemaLocation = anyURI > Content: (annotation?) </include> <redefine id = ID schemaLocation = anyURI> Content: (annotation (simpleType complexType group attributeGroup))* </redefine> <import id = ID namespace = anyURI schemaLocation = anyURI> Content: (annotation?) </import>	\$4.2.1, 4.2.2, 4.2.3 pt1

4Simple Data Type Declaration <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>	\$3.14.2 pt1 and \$4.1.2 pt2
--	------------------------------

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>	\$4.3 pt2 <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>
--	--

5Complex Data Type Declaration <complexType id = ID abstract = boolean : false' block = ('#all' List of ('extension' 'restriction')) final = ('#all' List of ('extension' 'restriction')) mixed = boolean : false' name = NCName > Content: (annotation?, (simpleContent complexContent ((group all choice sequence)?, ((attribute attributeGroup)*, anyAttribute?)))) </complexType>	\$3.4.2 pt1
Simple Content <simpleContent id = ID> Content: (annotation?, (restriction extension)) </simpleContent> <restriction id = ID base = QName> Content: (annotation?, (simpleType?, (minExclusive minInclusive maxExclusive maxInclusive totalDigits fractionDigits length minLength maxLength enumeration whiteSpace pattern)*?), ((attribute attributeGroup)*, anyAttribute?)) </restriction>	\$3.4.2 pt1
<extension id = ID base = QName> Content: (annotation?, ((attribute attributeGroup)*, anyAttribute?)) </extension>	
Complex Content <complexContent id = ID mixed = boolean> Content: (annotation?, (restriction extension)) </complexContent> <restriction id = ID base = QName> Content: (annotation?, (group all choice sequence)?, ((attribute attributeGroup)*, anyAttribute?)) </restriction>	\$3.4.2 pt1

<extension id = ID base = QName> Content: (annotation?, ((group all choice sequence)?, ((attribute attributeGroup)*, anyAttribute?))) </extension>	
6Element Declaration <element id = ID abstract = boolean : false' block = ('#all' List of ('extension' 'restriction' 'substitution')) default = string final = ('#all' List of ('extension' 'restriction')) fixed = string form = ('qualified' 'unqualified') maxOccurs = (nonNegativeInteger unbounded') : 1 minOccurs = nonNegativeInteger : 1 name = NCName nillable = boolean : false' ref = QName substitutionGroup = QName type = QName > Content: (annotation?, ((simpleType complexType)?, (unique key keyref*)) </element>	\$3.3.2 pt1
7Content Model <choice id = ID maxOccurs = (nonNegativeInteger unbounded') : 1 minOccurs = nonNegativeInteger : 1> Content: (annotation?, (element group choice sequence any)*) </choice> <sequence id = ID maxOccurs = (nonNegativeInteger unbounded') : 1 minOccurs = nonNegativeInteger : 1> Content: (annotation?, (element group choice sequence any)*) </sequence> <all id = ID maxOccurs = 1 : 1 minOccurs = (0 1) : 1> Content: (annotation?, element*) </all>	\$3.8.2 pt1
8Wildcard Schema Component <any id = ID maxOccurs = (nonNegativeInteger unbounded') : 1 minOccurs = nonNegativeInteger : 1 namespace = (('##any' '##other') List of (anyURI ('##targetNamespace' '##local'))) : '##any' processContents = ('lax' 'skip' 'strict') : 'strict' > Content: (annotation?) </any>	\$3.10.2 pt1
<anyAttribute id = ID namespace = (('##any' '##other') List of (anyURI ('##targetNamespace' '##local'))) : '##any' processContents = ('lax' 'skip' 'strict') : 'strict' > Content: (annotation?) </anyAttribute>	\$3.4.2 pt1
9Attribute Declaration <attribute id = ID default = string fixed = string form = ('qualified' 'unqualified') name = NCName ref = QName type = QName use = ('optional' 'prohibited' 'required') : 'optional' > Content: (annotation?, (simpleType?)) </attribute>	\$3.2.2 pt1

10	Element Group Declaration (<i>parameter entity like</i>)	§3.7.2 pt1
<pre><group id = ID maxOccurs = (nonNegativeInteger 'unbounded') : 1 minOccurs = nonNegativeInteger : 1 name = NCName ref = QName > Content: (annotation?, (all choice sequence?)) </group></pre>		

11	Attribute Group Declaration (<i>parameter entity like</i>)	§3.6.2 pt1
<pre><attributeGroup id = ID name = NCName ref = QName > Content: (annotation?, ((attribute attributeGroup)*, anyAttribute?)) </attributeGroup></pre>		

12	Identity-constraint Definitions	§3.11.2 pt1
<pre><unique id = ID name = NCName > Content: (annotation?, (selector, field+)) </unique></pre>		
<pre><key id = ID name = NCName > Content: (annotation?, (selector, field+)) </key></pre>		
<pre><keyref id = ID name = NCName refer = QName > Content: (annotation?, (selector, field+)) </keyref></pre>		
<pre><selector id = ID xpath = a subset of XPath expression > Content: (annotation?) </selector></pre>		
<pre><field id = ID xpath = a subset of XPath expression > Content: (annotation?) </field></pre>		

13	Schema Documentation Components	§3.13.2 pt1
<pre><annotation id = ID> Content: (appinfo documentation)* </annotation></pre>		
<pre><appinfo source = anyURI> Content: ({any})* </appinfo></pre>		
<pre><documentation source = anyURI xml:lang = language> Content: ({any})* </documentation></pre>		

14	Notation Declaration	§3.12.2 pt1
<pre><notation id = ID name = NCName public = anyURI system = anyURI > Content: (annotation?) </notation></pre>		

15	Defined Attribute Values	
{any}	Any element not part of Schema namespace.	
#all	All of the values listed	
	[final attribute] <i>controls further derivation</i>	§3.4.1 pt1
list	A finite-length (possibly empty) sequence of values	
union	A combination of the of one or more other datatypes.	
restriction	Values for constraining facets are specified to a subset of those	

	of its base type.	
	[namespace attribute] <i>controls use of namespaces</i>	§3.4.2 pt1
##any	Any namespace (default)	
##other	Any namespace other than target namespace	
##targetNamespace	Must belong to the target namespace of schema	
##local	Any unqualified XML from local namespace	
	[processContents attribute] <i>specify how contents should be processed for validation</i>	§3.10.1 pt1
strict	There must be a top-level declaration for the item available, or the item must have an xsi:type, and must be valid.	
skip	No constraints at all: the item must simply be well-formed.	
lax	Validate where you can, don't worry when you can't.	
	[form attribute] <i>controls namespace qualifying</i>	§3.2.2 pt1
qualified	Namespace qualified	
unqualified	No namespace qualification	
	[use attribute] <i>specifies the use of an attribute</i>	§3.2.2 pt1
optional	Attribute is optional	
prohibited	Attribute is prohibited	
required	Attribute is required to have a value	
	[whitespace attribute] <i>specifies whitespace handling</i>	§3.1.4 pt 1, §4.3.6 pt 2
preserve	The value is the normalized value	
replace	All occurrences of tab, line feed and carriage return are replaced with space.	
collapse	Contiguous sequences of spaces are collapsed to a single space, and initial and/or final spaces are deleted.	

16	Built-in Types	
anyType	Built-in Complex type definition of Ur-Type.	§3.4.7 pt1
anySimpleType	Built-in Simple type definition of Ur-Type.	§3.14.7 pt1

17	Schema Instance Related Markup	§2.6 pt1 and §3.2.7 pt1
xsi:type	An element in an instance may explicitly assert its type using the attribute xsi:type. The value is a QName associated with a type definition.	§2.6.1 pt1
xsi:nil	An element may be valid without content if it has the attribute xsi:nil with the value true.	§2.6.2 pt1
xsi:noNamespaceSchemaLocation, xsi:schemaLocation	Provide hints as to the physical location of schema documents	§2.6.3 pt1

18 Simple Data Types and Constraining Facets

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		

Simple Data Type	length	minLength	maxLength	pattern	enumeration	whiteSpace	maxInclusive	maxExclusive	minExclusive	minInclusive	totalDigits	fractionDigits
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

ver 1/03

