Advanced Web Programming

Schema Reference

Element	Explanation
all	Specifies that the child elements can appear in any order. Each child element can occur 0 or 1 time.
annotation	Specifies the top-level element for schema comments
any	Enables the author to extend the XML document with elements not specified by the schema
anyAttribute	Enables the author to extend the XML document with attributes not specified by the schema
appInfo	Specifies information to be used by the application (must go inside annotation)
attribute	Defines an attribute
attributeGroup	Defines an attribute group to be used in complex type definitions
choice	Allows only one of the elements contained in the <choice> declaration to be present within the containing element
complexContent	Defines extensions or restrictions on a complex type that contains mixed content or elements only
complexType	Defines a complex type element
documentation	Defines text comments in a schema (must go inside annotation)
element	Defines an element
extension	Extends an existing simpleType or complexType element
field	Specifies an XPath expression that specifies the value used to define an identity constraint
group	Defines a group of elements to be used in complex type definitions
import	Adds multiple schemas with different target namespace to a document
include	Adds multiple schemas with the same target namespace to a document
key	Specifies an attribute or element value as a key (unique, non-nullable, and always present) within the containing element in an instance document
keyref	Specifies that an attribute or element value correspond to those of the specified key or unique element
list	Defines a simple type element as a list of values
notation	Describes the format of non-XML data within an XML document
redefine	Redefines simple and complex types, groups, and attribute groups from an external schema
restriction	Defines restrictions on a simpleType, simpleContent, or a complexContent
schema	Defines the root element of a schema
selector	Specifies an XPath expression that selects a set of elements for an identity constraint
sequence	Specifies that the child elements must appear in a sequence. Each child element can occur from 0 to any number of times
simpleContent	Contains extensions or restrictions on a text-only complex type or on a simple type as content and contains no elements
simpleType	Defines a simple type and specifies the constraints and information about the values of attributes or text-only elements
union	Defines a simple type as a collection (union) of values from specified simple data types
unique	Defines that an element or an attribute value must be unique within the scope

Copyright © 2010 by Paul Roberts • All rights reserved

Data Type	Facets		Description
string	length pattern maxLength	minLength enumeration whiteSpace	Represents character strings.
Boolean	pattern	whiteSpace	Represents Boolean values, which are either true or false.
decimal	enumeration pattern totalDigits fractionDigits minInclusive	minExclusive maxInclusive maxExclusive whiteSpace	Represents arbitrary precision numbers.
float	pattern enumeration minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents single-precision 32-bit floating-point numbers.
double	pattern enumeration minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents double-precision 64-bit floating-point numbers.
duration	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a duration of time. The pattern for duration is PnYnMnDTnHnMnS, where nY represents the number of years, nM the number of months, nD the number of days, T the date/time separator, nH the number of hours, nM the number of minutes, and nS the number of seconds.
dateTime	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a specific instance of time. The pattern for dateTime is CCYY-MM-DDThh:mm:ss where CC represents the century, YY the year, MM the month, and DD the day, preceded by an optional leading negative (-) character to indicate a negative number. If the negative character is omitted, positive (+) is assumed. The T is the date/time separator and hh, mm, and ss represent hour, minute, and second respectively. Additional digits can be used to increase the precision of fractional seconds if desired. For example, the format ss.ss... with any number of digits after the decimal point is supported. The fractional seconds part is optional. This representation may be immediately followed by a "Z" to indicate Coordinated Universal Time (UTC) or to indicate the time zone. For example, the difference between the local time and Coordinated Universal Time, immediately followed by a sign, + or -, followed by the difference from UTC represented as hh:mm (minutes is required). If the time zone is included, both hours and minutes must be present.
time	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents an instance of time that recurs every day. The pattern for time is hh:mm:ss.sss with optional time zone indicator.
date	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a calendar date. The pattern for date is CCYY-MM-DD with optional time zone indicator as allowed for dateTime.
gYearMonth	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a specific Gregorian month in a specific Gregorian year. A set of one-month long, nonperiodic instances. The pattern for gYearMonth is CCYY-MM with optional time zone indicator.

Data Type	Facets		Description
gYear	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a Gregorian year. A set of one-year long, nonperiodic instances. The pattern for gYear is CCYY with optional time zone indicator as allowed for dateTime.
gMonthDay	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a specific Gregorian date that recurs, specifically a day of the year such as the third of May. A gMonthDay is the set of calendar dates. Specifically, it is a set of one-day long, annually periodic instances. The pattern for gMonthDay is --MM-DD with optional time zone indicator as allowed for date.
gDay	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a Gregorian day that recurs, specifically a day of the month such as the fifth day of the month. A gDay is the space of a set of calendar dates. Specifically, it is a set of one-day long, monthly periodic instances. The pattern for gDay is ---DD with optional time zone indicator as allowed for date.
gMonth	enumeration pattern minInclusive minExclusive	maxInclusive maxExclusive whiteSpace	Represents a Gregorian month that recurs every year. A gMonth is the space of a set of calendar months. Specifically, it is a set of one-month long, yearly periodic instances. The pattern for gMonth is --MM-- with optional time zone indicator as allowed for date.
hexBinary	length pattern maxLength	minLength enumeration whiteSpace	Represents arbitrary hex-encoded binary data. A hexBinary is the set of finite-length sequences of binary octets. Each binary octet is encoded as a character tuple, consisting of two hexadecimal digits ([0-9a-fA-F]) representing the octet code.
base64Binary	length pattern maxLength	minLength enumeration whiteSpace	Represents Base64-encoded arbitrary binary data. A base64Binary is the set of finite-length sequences of binary octets.
anyURI	length pattern maxLength	minLength enumeration whiteSpace	Represents a URI as defined by RFC 2396. An anyURI value can be absolute or relative, and may have an optional fragment identifier.
QName	length enumeration pattern	maxLength minLength whiteSpace	Represents a qualified name. A qualified name is composed of a prefix and a local name separated by a colon. Both the prefix and local names must be an NCName. The prefix must be associated with a namespace URI reference, using a namespace declaration.
NOTATION	length enumeration pattern	maxLength minLength whiteSpace	Represents a NOTATION attribute type. A set of QNames.

Data Type Facets

Simple types (both built-in and derived) have facets. A facet is a single defining aspect that helps determine the set of values for a simple type. For example, length, minInclusive, and maxInclusive are common facets for the built-in data types. All of the facets for a simple type define the set of legal values for that simple type.

A facet is defined as an element. Each facet element has a fixed attribute that is a Boolean value. When a simple type is defined, you can prevent derivatives of that type from modifying the value of specified facets. To prevent modification of a facet, add the fixed attribute to the facet and set its value to true.

Facets can only appear once in a type definition except for enumeration and pattern facets. Enumeration and pattern facets can have multiple entries and are grouped together.

Example

The following example shows a simple type with the fixed attribute set to true which prevents the length from having a value other than 7.

```
<xs:simpleType name="Postcode">
   <xs:restriction base="xs:string">
      <xs:length value="7" fixed="true"/>
   </xs:restriction>
</xs:simpleType>
```

List of Constraining Facets

The constraining facets (facets that can be used to constrain the values of simple types), their descriptions, and the built-in data types that they apply to, are listed below.

enumeration

Specified set of values. This constrains a data type to the specified values. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-enumeration.

fractionDigits

Value with specific maximum number of decimal digits in the fractional part. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-fractionDigits.

length

Number of units of length. Units of length depend on the data type. This value must be a nonNegativeInteger. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-length.

maxExclusive

Upper bound value (all values are less than this value). This value must be the same data type as the inherited data type. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-maxExclusive.

maxInclusive

Maximum value. This value must be the same data type as the inherited data type. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-maxInclusive.

maxLength

Maximum number of units of length. Units of length depend on the data type. This value must be a nonNegativeInteger. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-maxLength.

minExclusive

Lower bound value (all values are greater than this value). This value must be the same data type as the inherited data type. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-minExclusive.

minInclusive

Minimum value. This value must be the same data type as the inherited data type. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-minInclusive.

minLength

Minimum number of units of length. Units of length depend on the data type. This value must be a nonNegativeInteger. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-minLength.

pattern

Specific pattern that the data type's values must match. This constrains the data type to literals that match the specified pattern. The pattern value must be a regular expression. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-pattern.

totalDigits

Value with specific maximum number of decimal digits. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-totalDigits.

whiteSpace

Value must be one of preserve, replace, or collapse. The whiteSpace facet cannot be changed for most numeric data types. For more information, see the W3C XML Schema Part 2: Datatypes Recommendation at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#element-whiteSpace.