

XML Schema for W3C XML Schema Definition Language (XSD)

Contents

Overview	5
Namespace Summary	5
Schema File Summary	5
Namespace {http://www.w3.org/2001/XMLSchema}	6
Top-Level Elements (root element candidates)	7
<xs:schema>	7
Other Elements (global + local unified by type)	12
<xs:all>	12
<xs:all> (in xs:group)	15
<xs:annotation>	17
<xs:any>	20
<xs:anyAttribute>	23
<xs:appinfo>	25
<xs:attribute>	27
<xs:attribute> (type xs:attribute)	29
<xs:attributeGroup>	32
<xs:attributeGroup> (type xs:attributeGroupRef)	34
<xs:choice>	36
<xs:choice> (in xs:group)	39
<xs:complexContent>	41
<xs:complexType>	44
<xs:complexType> (type xs:localComplexType)	48
<xs:documentation>	51
<xs:element>	53
<xs:element> (type xs:localElement)	57
<xs:element> (type xs:narrowMaxMin)	61
<xs:enumeration>	65
<xs:extension> (in xs:complexContent)	67
<xs:extension> (in xs:simpleContent)	70
<xs:field>	72
<xs:fractionDigits>	75
<xs:group>	77
<xs:group> (type xs:groupRef)	79
<xs:import>	81
<xs:include>	83
<xs:key>	85
<xs:keyref>	87
<xs:length>	90
<xs:list>	92
<xs:maxExclusive>	94
<xs:maxInclusive>	96
<xs:maxLength>	98
<xs:minExclusive>	100
<xs:minInclusive>	102
<xs:minLength>	104
<xs:notation>	106
<xs:pattern>	109
<xs:redefine>	111
<xs:restriction>	114
<xs:restriction> (in xs:complexContent)	118
<xs:restriction> (in xs:simpleContent)	121
<xs:selector>	125



<xs:sequence>	128
<xs:sequence> (in xs:group).....	131
<xs:simpleContent>	133
<xs:simpleType>	135
<xs:simpleType> (type xs:localSimpleType).....	138
<xs:totalDigits>	140
<xs:union>.....	142
<xs:unique>.....	145
<xs:whiteSpace>	147
Complex Types	150
xs:all	150
xs:annotated.....	153
xs:anyType	156
xs:attribute	158
xs:attributeGroup.....	162
xs:attributeGroupRef	165
xs:complexRestrictionType	167
xs:complexType	170
xs:element.....	174
xs:explicitGroup	179
xs:extensionType.....	183
xs:facet	186
xs:group	188
xs:groupRef	192
xs:keybase	195
xs:localComplexType.....	197
xs:localElement	201
xs:localSimpleType	206
xs:namedAttributeGroup	209
xs:namedGroup	212
xs:narrowMaxMin	215
xs:noFixedFacet.....	220
xs:numFacet	222
xs:openAttrs	224
xs:realGroup	226
xs:restrictionType	229
xs:simpleExplicitGroup.....	233
xs:simpleExtensionType.....	236
xs:simpleRestrictionType	239
xs:simpleType	243
xs:topLevelAttribute.....	246
xs:topLevelComplexType	249
xs:topLevelElement.....	253
xs:topLevelSimpleType.....	257
xs:wildcard	260
Simple Types	263
xs:allNNI	263
xs:anyURI.....	265
xs:base64Binary	267
xs:blockSet	268
xs:boolean.....	270
xs:byte	272
xs:date.....	273
xs:dateTime	274
xs:decimal.....	275
xs:derivationControl	277
xs:derivationSet	279
xs:double	281
xs:duration	282
xs:ENTITIES	283
xs:ENTITY	284
xs:float.....	285
xs:formChoice	286

xs:fullDerivationSet.....	287
xs:gDay.....	289
xs:gMonth.....	290
xs:gMonthDay.....	291
xs:gYear.....	292
xs:gYearMonth.....	293
xs:hexBinary.....	294
xs:ID.....	295
xs:IDREF.....	296
xs:IDREFS.....	297
xs:int.....	298
xs:integer.....	299
xs:language.....	301
xs:long.....	303
xs:Name.....	304
xs:namespaceList.....	306
xs:NCName.....	308
xs:negativeInteger.....	310
xs:NMTOKEN.....	311
xs:NMTOKENS.....	313
xs:nonNegativeInteger.....	314
xs:nonPositiveInteger.....	316
xs:normalizedString.....	317
xs:NOTATION.....	319
xs:positiveInteger.....	320
xs:public.....	321
xs:QName.....	322
xs:reducedDerivationControl.....	324
xs:short.....	325
xs:simpleDerivationSet.....	326
xs:string.....	328
xs:time.....	330
xs:token.....	331
xs:typeDerivationControl.....	333
xs:unsignedByte.....	334
xs:unsignedInt.....	335
xs:unsignedLong.....	336
xs:unsignedShort.....	337
Element Groups.....	338
xs:allModel.....	338
xs:attrDecls.....	340
xs:complexTypeModel.....	342
xs:facets.....	344
xs:identityConstraint.....	347
xs:nestedParticle.....	349
xs:particle.....	351
xs:redefinable.....	353
xs:schemaTop.....	355
xs:simpleDerivation.....	357
xs:simpleRestrictionModel.....	358
xs:typeDefParticle.....	361
Attribute Groups.....	363
xs:defRef.....	363
xs:occurs.....	365
Namespace {http://www.w3.org/XML/1998/namespace}.....	367
Global Attributes.....	368
xml:base.....	368
xml:id.....	369
xml:lang.....	370
xml:space.....	372
Attribute Groups.....	374
xml:specialAttrs.....	374
Schema "xml.xsd".....	376

Global Attribute Summary	377
Attribute Group Summary	377
Schema XML Source	378
Schema "XMLSchema.xsd"	383
Top-Level Element Summary (root element candidates).....	384
Other Element Summary (global + local unified by type)	384
Complex Type Summary	389
Simple Type Summary	392
Element Group Summary	395
Attribute Group Summary	396
Schema XML Source	396
Namespace Bindings	428

Overview

Namespace Summary		Page
<p>{} http://www.w3.org/2001/XMLSchema</p> <p>XML Schemas (1): XMLSchema.xsd XSD Components: elements (41 global + 28 local), complexTypes (35), simpleTypes (55), element groups (12), attribute groups (2)</p>		6
<p>{} http://www.w3.org/XML/1998/namespace</p> <p>XML Schemas (1): xml.xsd XSD Components: global attributes (4), attribute groups (1)</p>		367

Schema File Summary		Page
<p> xml.xsd</p>	<p>About the XML namespace This schema document describes the XML namespace, in a form suitable for import by other schema documents.</p> <p>Schema Source: http://www.w3.org/2001/xml.xsd; see XML source [378]</p> <p>Target Namespace: http://www.w3.org/XML/1998/namespace</p> <p>Defined Components: global attributes (4), attribute groups (1)</p> <p>Default Namespace-Qualified Form: Local Elements: unqualified; Local Attributes: unqualified</p> <p>Imported by Schemas (1): XMLSchema.xsd</p>	376
<p> XMLSchema.xsd</p>	<p>Part 1 version: Id: structures.xsd,v 1.2 2004/01/15 11:34:25 ht Exp Part 2 version: Id: datatypes.xsd,v 1.3 2004/01/23 18:11:13 ht Exp</p> <p>Schema Source: http://www.w3.org/2001/XMLSchema.xsd; see XML source [396]</p> <p>Target Namespace: http://www.w3.org/2001/XMLSchema</p> <p>Version: 1.0</p> <p>Defined Components: elements (top-level / other; 41 global + 28 local), complexTypes (35), simpleTypes (55), element groups (12), attribute groups (2)</p> <p>Default Namespace-Qualified Form: Local Elements: qualified; Local Attributes: unqualified</p> <p>Default Block Attribute: "#all" (<i>blocks all substitutions of elements and their types both through substitution groups and xsi:type attribute in instance XML documents</i>)</p> <p>Imports Schemas (1): xml.xsd</p>	383

Namespace {http://www.w3.org/2001/XMLSchema}

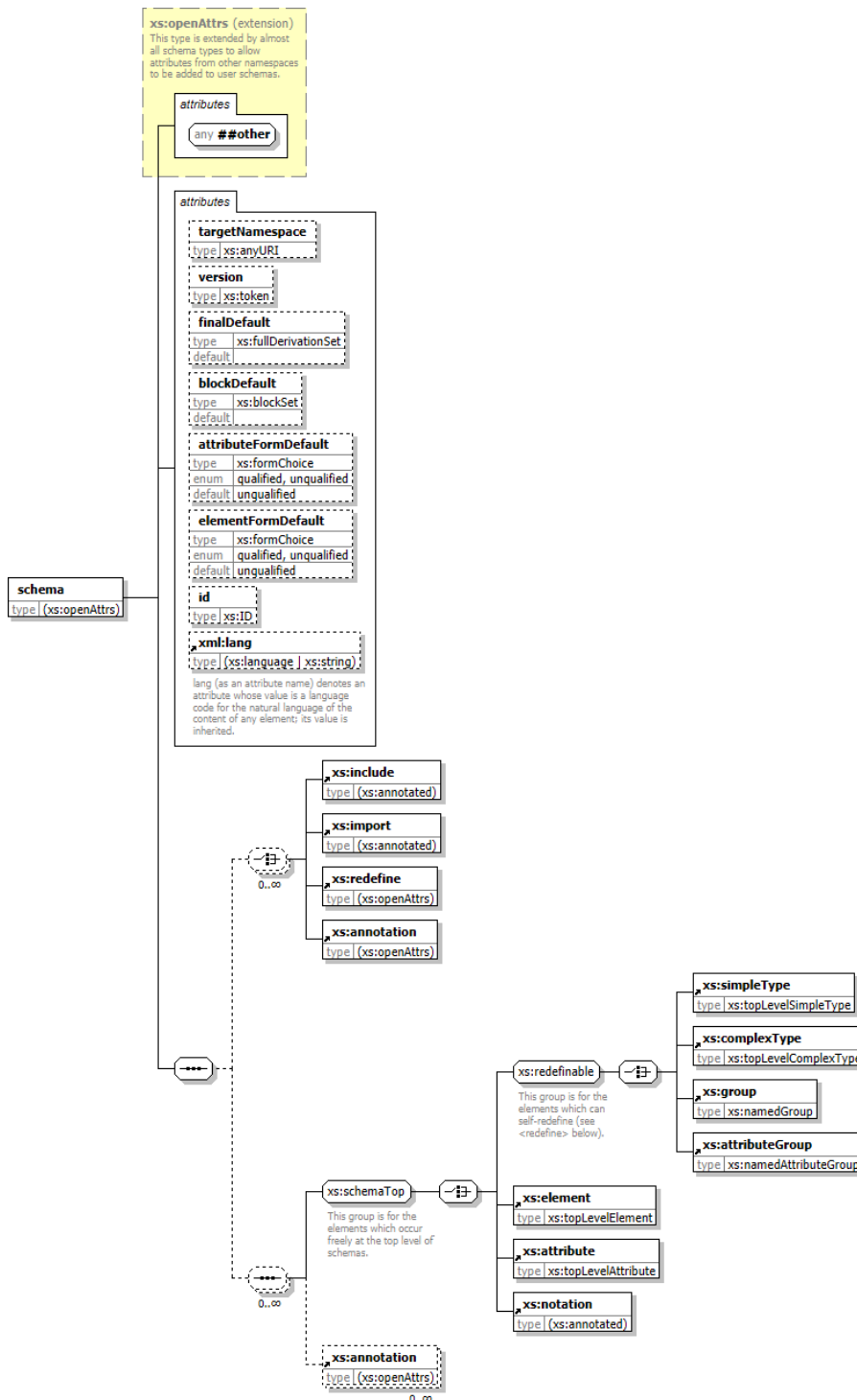
Top-Level Elements (root element candidates)

global element

<xs:schema>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of `xs:openAttrs`) [8]
Content: complex, 8 attributes, attr. wildcard, 12 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in `XMLSchema.xsd`; see [XML source](#) [8]

Component Diagram



XML Representation Summary

```

<xs:schema
  targetNamespace      = xs:anyURI
  version              = xs:token
  finalDefault         = ("#all" | list of ("extension" | "restriction" | "list" | "union")) : ""
  blockDefault        = ("#all" | list of ("extension" | "restriction" | "substitution")) : ""
  attributeFormDefault = ("qualified" | "unqualified") : "unqualified"
  elementFormDefault  = ("qualified" | "unqualified") : "unqualified"
  id                  = xs:ID
  xml:lang             = (xs:language | "")
  {any attribute from non-schema namespace}
>
Content: (xs:include | xs:import | xs:redefine | xs:annotation)*, ((xs:simpleType | xs:complexType |
  xs:group | xs:attributeGroup | xs:element | xs:attribute | xs:notation), xs:annotation)*
</xs:schema>

```

Content model elements (11):

xs:annotation [17], xs:attribute [27], xs:attributeGroup [32], xs:complexType [44], xs:element [53],
 xs:group [77], xs:import [81], xs:include [83], xs:notation [106], xs:redefine [111], xs:simpleType [135]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-schema>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── complexType

```

XML Source (see within schema source: p. 399)

```

<xs:element id="schema" name="schema">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-schema"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:sequence>
          <xs:choice maxOccurs="unbounded" minOccurs="0">
            <xs:element ref="xs:include"/>
            <xs:element ref="xs:import"/>
            <xs:element ref="xs:redefine"/>
            <xs:element ref="xs:annotation"/>
          </xs:choice>
          <xs:sequence maxOccurs="unbounded" minOccurs="0">
            <xs:group ref="xs:schemaTop"/>
            <xs:element maxOccurs="unbounded" minOccurs="0" ref="xs:annotation"/>
          </xs:sequence>
        </xs:sequence>
        <xs:attribute name="targetNamespace" type="xs:anyURI"/>
        <xs:attribute name="version" type="xs:token"/>
        <xs:attribute default="" name="finalDefault" type="xs:fullDerivationSet" use="optional"/>
        <xs:attribute default="" name="blockDefault" type="xs:blockSet" use="optional"/>
        <xs:attribute default="unqualified" name="attributeFormDefault" type="xs:formChoice" use="optional"/>
        <xs:attribute default="unqualified" name="elementFormDefault" type="xs:formChoice" use="optional"/>
        <xs:attribute name="id" type="xs:ID"/>
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:key name="element">
    <xs:selector xpath="xs:element"/>

```

```

    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="attribute">
    <xs:selector xpath="xs:attribute"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="type">
    <xs:selector xpath="xs:complexType|xs:simpleType"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="group">
    <xs:selector xpath="xs:group"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="attributeGroup">
    <xs:selector xpath="xs:attributeGroup"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="notation">
    <xs:selector xpath="xs:notation"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="identityConstraint">
    <xs:selector xpath="./xs:key|./xs:unique|./xs:keyref"/>
    <xs:field xpath="@name"/>
  </xs:key>
</xs:element>

```

Attribute Detail (all declarations; 9/9)

attributeFormDefault

Type: [xs:formChoice](#) [286]
Use: optional
Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"
Default: "unqualified"

blockDefault

Type: [xs:blockSet](#) [268]
Use: optional
Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "substitution")
```

Default: ""

elementFormDefault

Type: [xs:formChoice](#) [286]
Use: optional
Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"
Default: "unqualified"

finalDefault

Type: [xs:fullDerivationSet](#) [287]
Use: optional

Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "list" | "union")
```

Default: ""

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

■ targetNamespace

Type: [xs:anyURI](#) [265]

Use: optional

Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

■ version

Type: [xs:token](#) [331]

Use: optional

Defined: locally within (this) [xs:schema](#) element; see [XML source](#) [399]

■ xml:lang [370]

Type: [anonymous](#) simpleType ([union of](#) ([xs:language](#) | [restriction of xs:string](#))) [370]

Use: optional

Defined: by reference within (this) [xs:schema](#) element; see [XML source](#) [399]

Attribute Value

```
xs:language | ""
```

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 12/12)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content

Defined: by reference within (this) [xs:schema](#) element; see [XML source](#) [399]

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content

Defined: by reference within (this) [xs:schema](#) element; see [XML source](#) [399]

↔ [xs:attribute](#) [27]

Type: [xs:topLevelAttribute](#) [246], complex content

Defined: [by reference](#) [356] within [xs:schemaTop](#) group; see [XML source](#) [398]

↔ [xs:attributeGroup](#) [32]

Type: [xs:namedAttributeGroup](#) [209], complex content

Defined: [by reference](#) [354] within [xs:redefinable](#) group; see [XML source](#) [398]

↔ [xs:complexType](#) [44]

Type: [xs:topLevelComplexType](#) [249], complex content

Defined: [by reference](#) [354] within [xs:redefinable](#) group; see [XML source](#) [398]

↔ [xs:element](#) [53]

Type: [xs:topLevelElement](#) [253], complex content
Defined: [by reference](#) [356] within [xs:schemaTop](#) group; see [XML source](#) [398]

↔ [xs:group](#) [77]

Type: [xs:namedGroup](#) [212], complex content
Defined: [by reference](#) [354] within [xs:redefinable](#) group; see [XML source](#) [398]

↔ [xs:import](#) [81]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [82], complex content
Defined: [by reference](#) within ([this](#)) [xs:schema](#) element; see [XML source](#) [399]

↔ [xs:include](#) [83]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [84], complex content
Defined: [by reference](#) within ([this](#)) [xs:schema](#) element; see [XML source](#) [399]

↔ [xs:notation](#) [106]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [107], complex content
Defined: [by reference](#) [356] within [xs:schemaTop](#) group; see [XML source](#) [398]

↔ [xs:redefine](#) [111]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [112], complex content
Defined: [by reference](#) within ([this](#)) [xs:schema](#) element; see [XML source](#) [399]

↔ [xs:simpleType](#) [135]

Type: [xs:topLevelSimpleType](#) [257], complex content
Defined: [by reference](#) [354] within [xs:redefinable](#) group; see [XML source](#) [398]

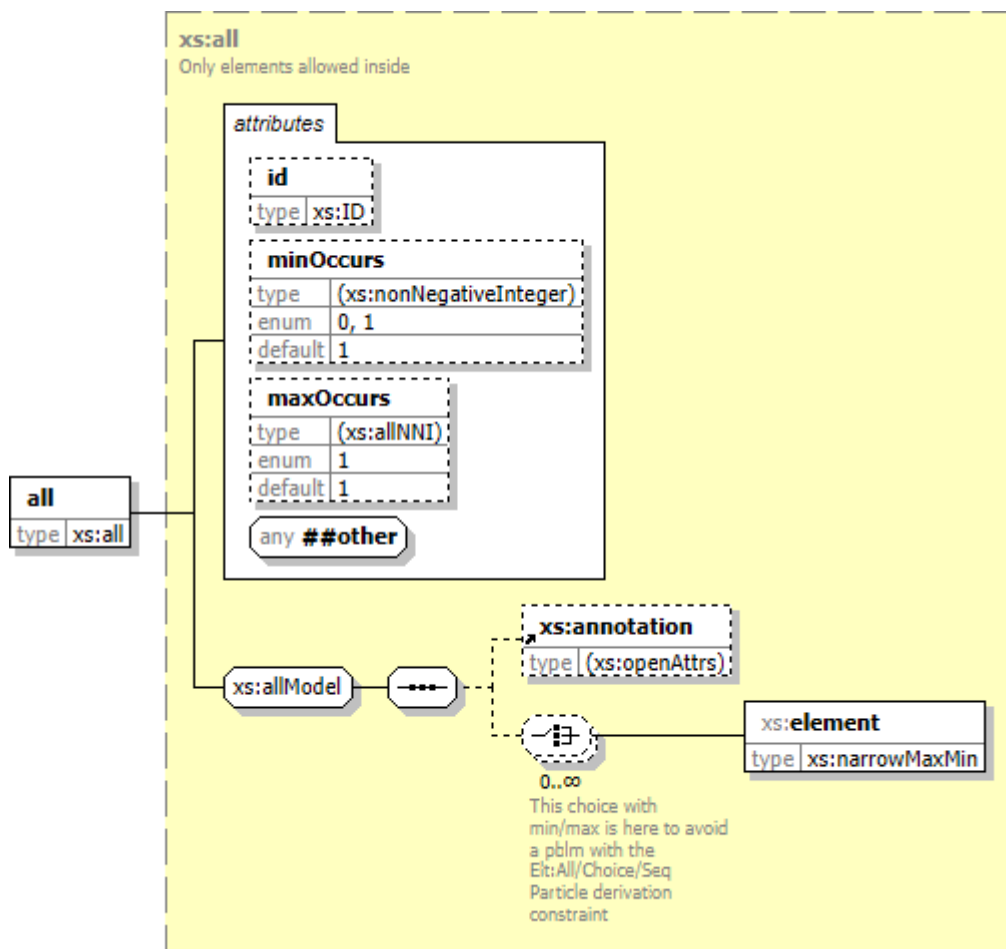
Other Elements (global + local unified by type)

global element

<xs:all>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:all](#) [150]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 2 [elements](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [13]

Component Diagram



XML Representation Summary

```
<xs:all
  id           = xs:ID
  minOccurs    = ("0" | "1") : "1"
  maxOccurs    = "1" : "1"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:element*
</xs:all>
```

Content model elements (2):

[xs:annotation](#) [17], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

Included in content model of elements (4):

[xs:complexType](#) [44], [xs:extension](#) (in [xs:complexContent](#)) [67],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:restriction](#) (in [xs:complexContent](#)) [118]

Known Usage Locations

- **Within global complexTypes (1):**
[xs:realGroup](#) [228]
- **Within model groups (2):**
[xs:particle](#) [351], [xs:typeDefParticle](#) [362]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-all>

XML Source (see within schema source: p. 408)

```
<xs:element id="all" name="all" type="xs:all">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-all"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs

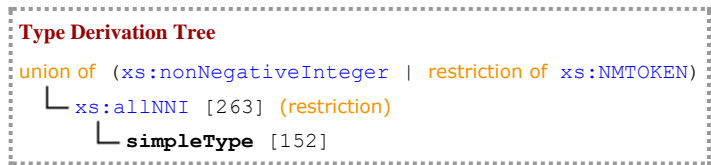
Type: [anonymous simpleType](#) ([restriction of xs:allNNI](#)) [13]
Use: optional
Defined: [locally](#) [152] within [xs:all](#) complexType; see [XML source](#) [408]

Attribute Value

enumeration of ([xs:nonNegativeInteger](#) | "unbounded")

Enumeration: "1"
Default: "1"

Anonymous simpleType



■ minOccurs

Type: [anonymous simpleType](#) ([restriction of xs:nonNegativeInteger](#)) [14]
Use: optional
Defined: [locally](#) [152] within [xs:all](#) complexType; see [XML source](#) [407]

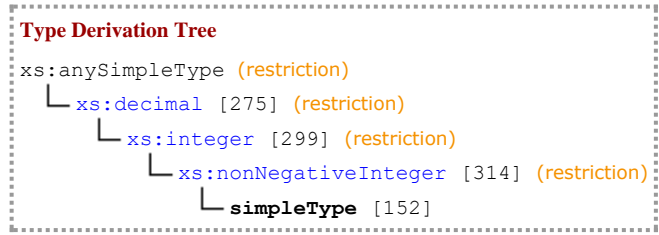
Attribute Value

enumeration of [xs:nonNegativeInteger](#)

Enumeration: "0", "1"

Default: "1"

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: locally [152] within xs:all complexType; see XML source [408]

Content Element Detail (all declarations; 2/2)

↔ xs:annotation [17]

Type: anonymous complexType (extension of xs:openAttrs) [18], complex content
Defined: by reference [339] within xs:allModel group; see XML source [407]

↔ xs:element [61]

Type: xs:narrowMaxMin [215], complex content
Defined: locally [339] within xs:allModel group; see XML source [407]

local element

<xs:all> (in xs:group)

Namespace: <http://www.w3.org/2001/XMLSchema>

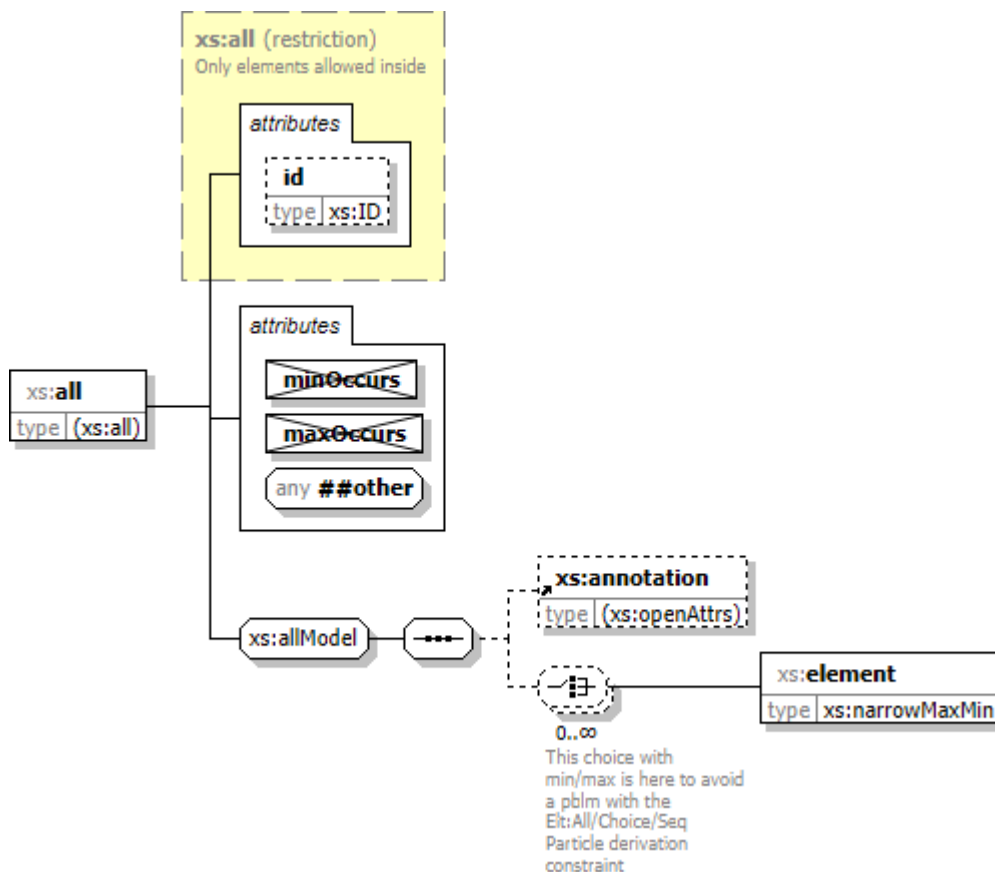
Type: anonymous complexType (restriction of [xs:all](#)) [16]

Content: complex, 1 attribute, attr. wildcard, 2 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within [xs:namedGroup](#) complexType [214] in [XMLSchema.xsd](#); see [XML source](#) [16]

Component Diagram



XML Representation Summary

```
<xs:all
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:element*
</xs:all>
```

Content model elements (2):

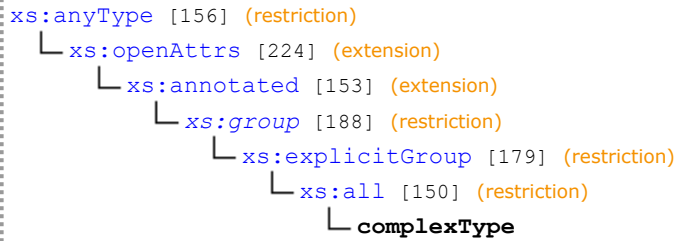
[xs:annotation](#) [17], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

Included in content model of elements (1):

[xs:group](#) [77]

Anonymous Type Detail

Type Derivation Tree



XML Source (see within schema source: p. 406)

```

<xs:element name="all">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:all">
        <xs:group ref="xs:allModel"/>
        <xs:attribute name="minOccurs" use="prohibited"/>
        <xs:attribute name="maxOccurs" use="prohibited"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 4/4)

- id
 - Type: [xs:ID](#) [295]
 - Use: optional
 - Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]
- maxOccurs
 - Use: prohibited
- minOccurs
 - Use: prohibited
- {any attribute from non-schema namespace}
 - Defined: locally within (this) [xs:all](#) element; see [XML source](#) [406]

Content Element Detail (all declarations; 2/2)

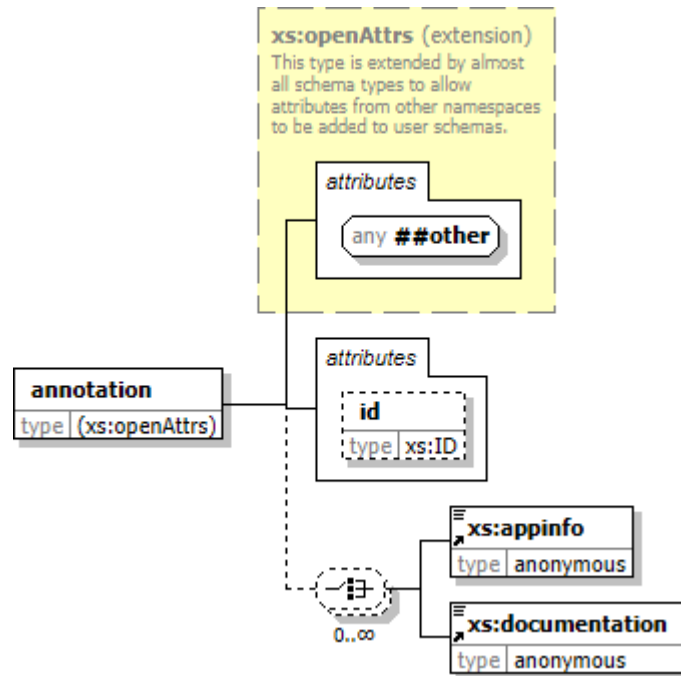
- [xs:annotation](#) [17]
 - Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content
 - Defined: by reference [339] within [xs:allModel](#) group; see [XML source](#) [407]
- [xs:element](#) [61]
 - Type: [xs:narrowMaxMin](#) [215], complex content
 - Defined: locally [339] within [xs:allModel](#) group; see [XML source](#) [407]

global element

<xs:annotation>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:openAttrs](#)) [18]
Content: complex, 1 attribute, attr. wildcard, 2 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [18]

Component Diagram



XML Representation Summary

```

<xs:annotation
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: (xs:appinfo | xs:documentation)*
</xs:annotation>
    
```

Content model elements (2):

[xs:appinfo](#) [25], [xs:documentation](#) [51]

Included in content model of elements (52):

[xs:all](#) [12],
[xs:all](#) (in [xs:group](#)) [15],
[xs:any](#) [20],
[xs:anyAttribute](#) [23],
[xs:attribute](#) [27],
[xs:attribute](#) (type [xs:attribute](#)) [29],
[xs:attributeGroup](#) [32],
[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34],
[xs:choice](#) [36],
[xs:choice](#) (in [xs:group](#)) [39],
[xs:complexContent](#) [41],
[xs:complexType](#) [44],
[xs:complexType](#) (type [xs:localComplexType](#)) [48],
[xs:element](#) [53],
[xs:keyref](#) [87],
[xs:length](#) [90],
[xs:list](#) [92],
[xs:maxExclusive](#) [94],
[xs:maxInclusive](#) [96],
[xs:maxLength](#) [98],
[xs:minExclusive](#) [100],
[xs:minInclusive](#) [102],
[xs:minLength](#) [104],
[xs:notation](#) [106],
[xs:pattern](#) [109],
[xs:redefine](#) [111],
[xs:restriction](#) [114],
[xs:restriction](#) (in [xs:complexContent](#)) [118],

[xs:element](#) (type [xs:localElement](#)) [57],
[xs:element](#) (type [xs:narrowMaxMin](#)) [61],
[xs:enumeration](#) [65],
[xs:extension](#) (in [xs:complexContent](#)) [67],
[xs:extension](#) (in [xs:simpleContent](#)) [70],
[xs:field](#) [72],
[xs:fractionDigits](#) [75],
[xs:group](#) [77],
[xs:group](#) (type [xs:groupRef](#)) [79],
[xs:import](#) [81],
[xs:include](#) [83],
[xs:key](#) [85],
[xs:restriction](#) (in [xs:simpleContent](#)) [121],
[xs:schema](#) [7],
[xs:selector](#) [125],
[xs:sequence](#) [128],
[xs:sequence](#) (in [xs:group](#)) [131],
[xs:simpleContent](#) [133],
[xs:simpleType](#) [135],
[xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
[xs:totalDigits](#) [140],
[xs:union](#) [142],
[xs:unique](#) [145],
[xs:whiteSpace](#) [147]

Known Usage Locations

- **Within global complexTypes (21):**

[xs:annotated](#) [155], [xs:attributeGroupRef](#) [166], [xs:complexRestrictionType](#) [169], [xs:explicitGroup](#) [181],
[xs:groupRef](#) [194], [xs:localComplexType](#) [199], [xs:localElement](#) [204], [xs:localSimpleType](#) [207],
[xs:namedAttributeGroup](#) [210], [xs:namedGroup](#) [214], [xs:narrowMaxMin](#) [219], [xs:noFixedFacet](#) [221],
[xs:numFacet](#) [223], [xs:realGroup](#) [228], [xs:simpleExplicitGroup](#) [234], [xs:simpleExtensionType](#) [237],
[xs:simpleRestrictionType](#) [241], [xs:topLevelAttribute](#) [248], [xs:topLevelComplexType](#) [251],
[xs:topLevelElement](#) [256], [xs:topLevelSimpleType](#) [259]

- **Within anonymous complexTypes of elements (6):**

[xs:pattern](#) [110], [xs:redefine](#) [112], [xs:schema](#) [10], [xs:schema](#) [10], [xs:totalDigits](#) [141], [xs:whiteSpace](#) [149]

- **Within model groups (1):**

[xs:allModel](#) [339]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-annotation>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── complexType
    
```

XML Source (see within schema source: p. 413)

```

<xs:element id="annotation" name="annotation">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-annotation"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:choice maxOccurs="unbounded" minOccurs="0">
          <xs:element ref="xs:appinfo"/>
          <xs:element ref="xs:documentation"/>
        </xs:choice>
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 2/2)

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: locally within ([this](#)) [xs:annotation](#) element; see [XML source](#) [413]

■ *{any attribute from non-schema namespace}*

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 2/2)

↔ [xs:appinfo](#) [25]

Type: [anonymous](#) complexType [412], mixed content

Defined: by reference within ([this](#)) [xs:annotation](#) element; see [XML source](#) [413]

↔ [xs:documentation](#) [51]

Type: [anonymous](#) complexType [412], mixed content

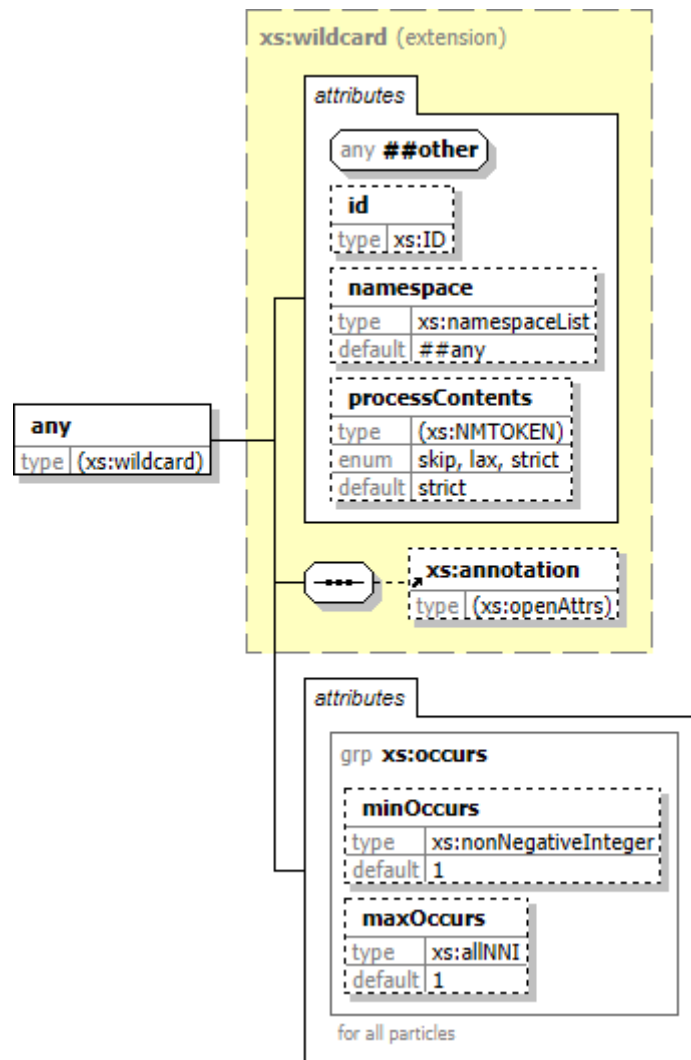
Defined: by reference within ([this](#)) [xs:annotation](#) element; see [XML source](#) [413]

global element

<xs:any>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: anonymous complexType (extension of [xs:wildcard](#)) [21]
 Content: complex, 5 attributes, attr. [wildcard](#), 1 element
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [21]

Component Diagram



XML Representation Summary

```
<xs:any
  id = xs:ID
  namespace = ( ("##any" | "##other" ) | list of (xs:anyURI | ("##targetNamespace" | "##local"))) :
  "##any"
  processContents = ("skip" | "lax" | "strict") : "strict"
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
  {any attribute from non-schema namespace}
  >
  Content: xs:annotation?
</xs:any>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (4):

[xs:choice](#) [36], [xs:sequence](#) [128],
[xs:choice](#) (in [xs:group](#)) [39], [xs:sequence](#) (in [xs:group](#)) [131]

Known Usage Locations

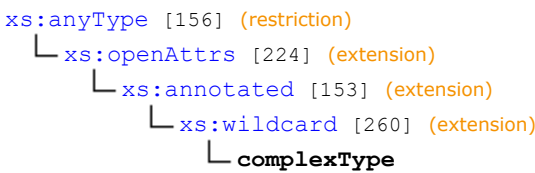
- Within model groups (2):
[xs:nestedParticle](#) [349], [xs:particle](#) [352]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-any>

Anonymous Type Detail

Type Derivation Tree



XML Source (see within schema source: p. 408)

```
<xs:element id="any" name="any">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-any"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:wildcard">
        <xs:attributeGroup ref="xs:occurs"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Attribute Detail (all declarations; 6/6)

id
Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs
Type: [xs:allNNI](#) [263]
Use: optional
Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

minOccurs
Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

■ namespace

Type: [xs:namespaceList](#) [306]

Use: optional

Defined: [locally](#) [261] within [xs:wildcard](#) complexType; see [XML source](#) [408]

Attribute Value

```

(“##any” | “##other”) | list of (xs:anyURI | (“##targetNamespace” | “##local”))
    
```

Default: “##any”

■ processContents

Type: [anonymous simpleType](#) ([restriction of xs:NMTOKEN](#)) [22]

Use: optional

Defined: [locally](#) [261] within [xs:wildcard](#) complexType; see [XML source](#) [408]

Attribute Value

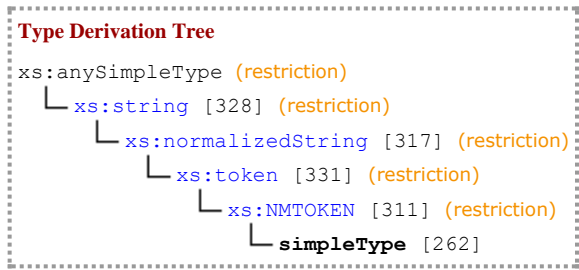
```

enumeration of xs:NMTOKEN
    
```

Enumeration: "skip", "lax", "strict"

Default: "strict"

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

Type: [anonymous complexType](#) ([extension of xs:openAttrs](#)) [18], complex content

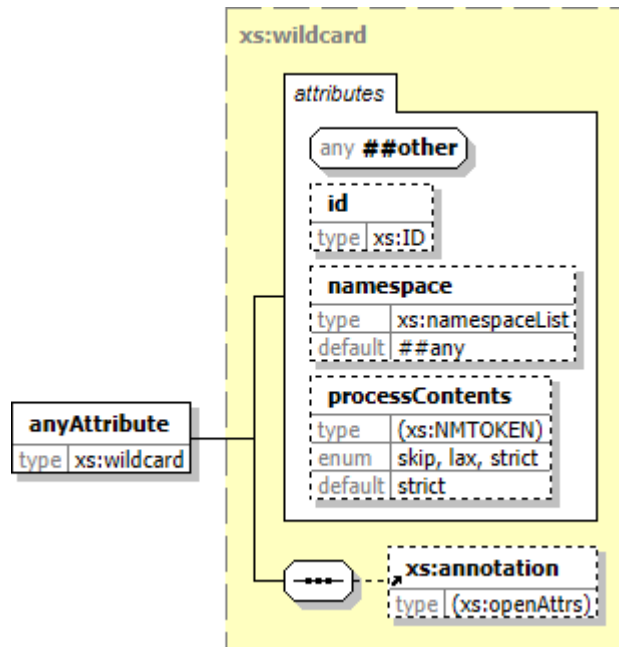
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:anyAttribute>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:wildcard](#) [260]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 1 [element](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [24]

Component Diagram



XML Representation Summary

```

<xs:anyAttribute
  id = xs:ID
  namespace = (("##any" | "##other") | list of (xs:anyURI | ("##targetNamespace" | "##local"))) :
  ##any"
  processContents = ("skip" | "lax" | "strict") : "strict"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:anyAttribute>
    
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (7):

[xs:attributeGroup](#) [32], [xs:extension](#) (in [xs:simpleContent](#)) [70],
[xs:complexType](#) [44], [xs:restriction](#) (in [xs:complexType](#)) [118],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:restriction](#) (in [xs:simpleContent](#)) [121]
[xs:extension](#) (in [xs:complexType](#)) [67],

Known Usage Locations

- Within model groups (1):

[xs:attrDecls](#) [340]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-anyAttribute>

XML Source (see within schema source: p. 401)

```
<xs:element id="anyAttribute" name="anyAttribute" type="xs:wildcard">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-anyAttribute"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

namespace

Type: [xs:namespaceList](#) [306]
Use: optional
Defined: [locally](#) [261] within [xs:wildcard](#) complexType; see [XML source](#) [408]

Attribute Value

```
("##any" | "##other") | list of (xs:anyURI | ("##targetNamespace" | "##local"))
```

Default: "##any"

processContents

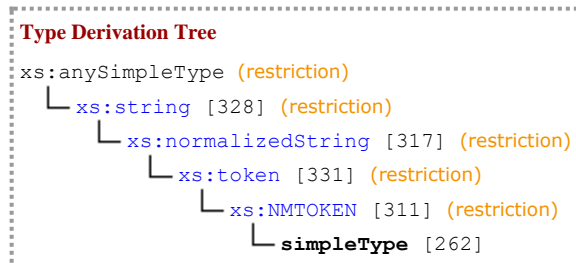
Type: [anonymous simpleType](#) ([restriction of xs:NMTOKEN](#)) [24]
Use: optional
Defined: [locally](#) [261] within [xs:wildcard](#) complexType; see [XML source](#) [408]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "skip", "lax", "strict"
Default: "strict"

Anonymous simpleType



{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

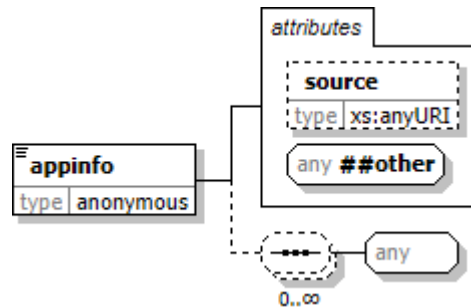
Type: [anonymous complexType](#) ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:appinfo>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [anonymous](#) complexType [412]
Content: mixed (*allows character data*), 1 [attribute](#), attr. [wildcard](#), elem. [wildcard](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [25]

Component Diagram



XML Representation Summary

```
<xs:appinfo
  source = xs:anyURI
  {any attribute from non-schema namespace}
>
Content: {text} × {any}*
</xs:appinfo>
```

Included in content model of elements (1):

[xs:annotation](#) [17]

Known Usage Locations

- Within anonymous complexTypes of elements (1):

[xs:annotation](#) [19]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-appinfo>

XML Source (see within schema source: p. 412)

```
<xs:element id="appinfo" name="appinfo">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-appinfo"/>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:sequence maxOccurs="unbounded" minOccurs="0">
      <xs:any processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="source" type="xs:anyURI"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:element>
```

Attribute Detail (all declarations; 2/2)

■ source

Type: [xs:anyURI](#) [265]

Use: optional

Defined: locally within ([this](#)) [xs:appinfo](#) element; see [XML source](#) [412]

■ {any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:appinfo](#) element; see [XML source](#) [412]

Content Element Detail (all declarations; 1/1)

↔ {any element from any namespace}

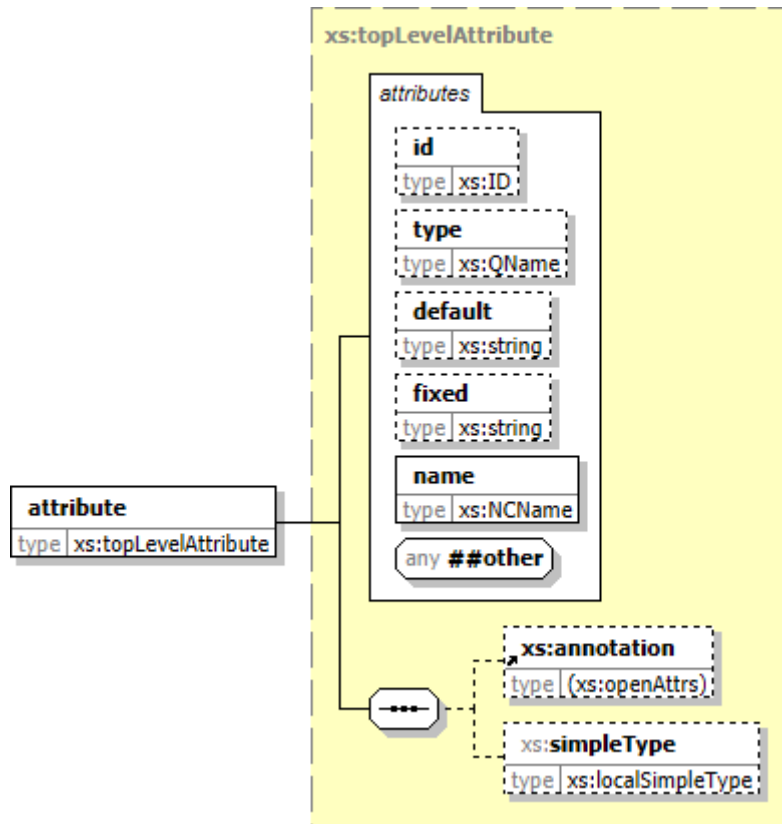
Defined: locally within ([this](#)) [xs:appinfo](#) element; see [XML source](#) [412]

global element

<xs:attribute>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:topLevelAttribute](#) [246]
Content: complex, 5 attributes, attr. wildcard, 2 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [28]

Component Diagram



XML Representation Summary

```

<xs:attribute
  id      = xs:ID
  type   = xs:QName
  default = xs:string
  fixed  = xs:string
  name   = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType?
</xs:attribute>
    
```

Content model elements (2):

[xs:annotation](#) [17], [xs:simpleType](#) (type `xs:localSimpleType`) [138]

Included in content model of elements (1):

[xs:schema](#) [7]

Known Usage Locations

- Within model groups (1):

[xs:schemaTop](#) [356]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-attribute>

XML Source (see within schema source: p. 409)

```
<xs:element id="attribute" name="attribute" type="xs:topLevelAttribute">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-attribute"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 6/6)

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [247] within [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

{any attribute from non-schema namespace}

Defined: [locally](#) [248] within [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

Content Element Detail (all declarations; 2/2)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [248] within [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) [248] within [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

local element

<xs:attribute> (type xs:attribute)

Namespace: <http://www.w3.org/2001/XMLSchema>

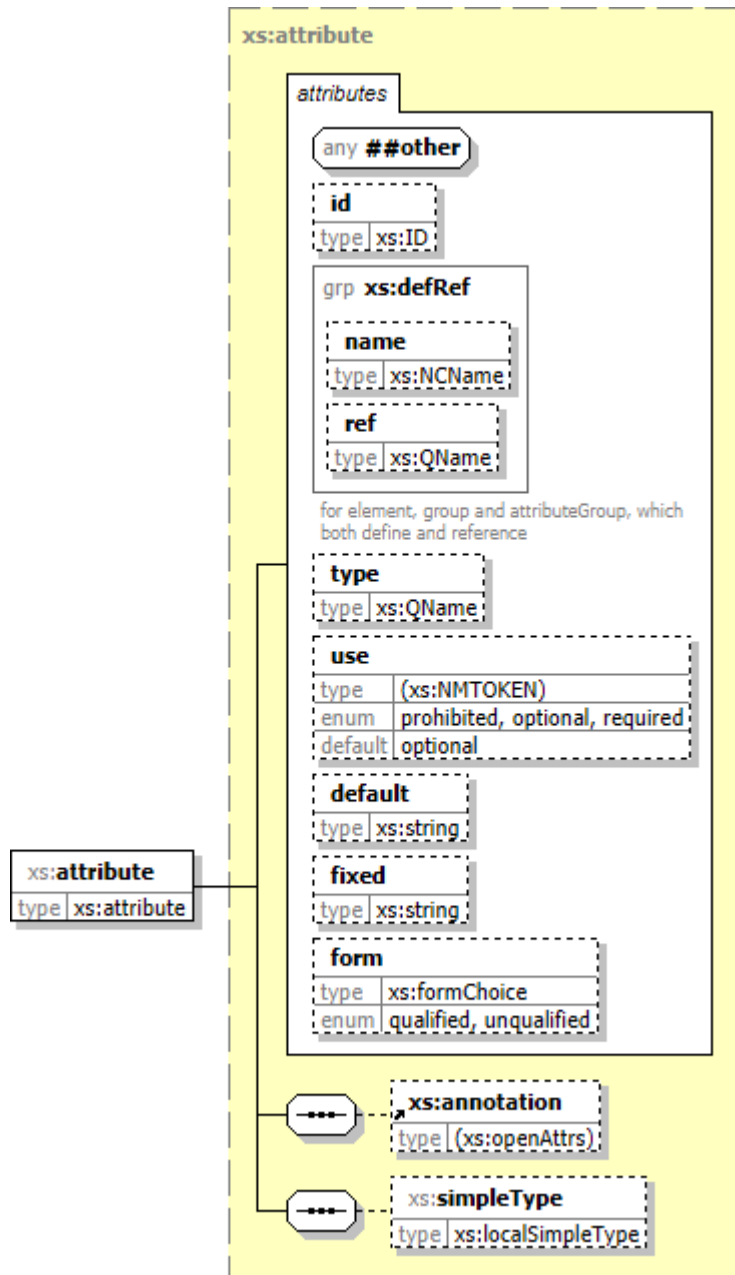
Type: [xs:attribute](#) [158]

Content: complex, 8 attributes, attr. wildcard, 2 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within [xs:attrDecls](#) group [340] in [XMLSchema.xsd](#); see [XML source](#) [30]

Component Diagram



XML Representation Summary

```
<xs:attribute
  id       = xs:ID
  name    = xs:NCName
  ref     = xs:QName
  type    = xs:QName
  use     = ("prohibited" | "optional" | "required") : "optional"
  default = xs:string
  fixed   = xs:string
  form    = ("qualified" | "unqualified")
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType?
</xs:attribute>
```

Content model elements (2):

[xs:annotation](#) [17], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Included in content model of elements (7):

[xs:attributeGroup](#) [32], [xs:extension](#) (in [xs:simpleContent](#)) [70],
[xs:complexType](#) [44], [xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:restriction](#) (in [xs:simpleContent](#)) [121]
[xs:extension](#) (in [xs:complexContent](#)) [67],

XML Source (see within schema source: p. 401)

```
<xs:element name="attribute" type="xs:attribute"/>
```

Attribute Detail (all declarations; 9/9)

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

form

Type: [xs:formChoice](#) [286]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: optional
Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ ref

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

■ use

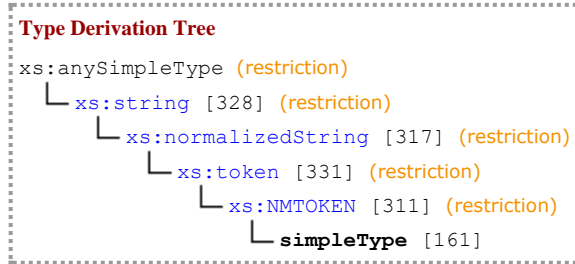
Type: [anonymous](#) simpleType ([restriction of xs:NMTOKEN](#)) [31]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

Attribute Value

`enumeration of xs:NMTOKEN`

Enumeration: "prohibited", "optional", "required"
Default: "optional"

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 2/2)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:simpleType](#) [138]

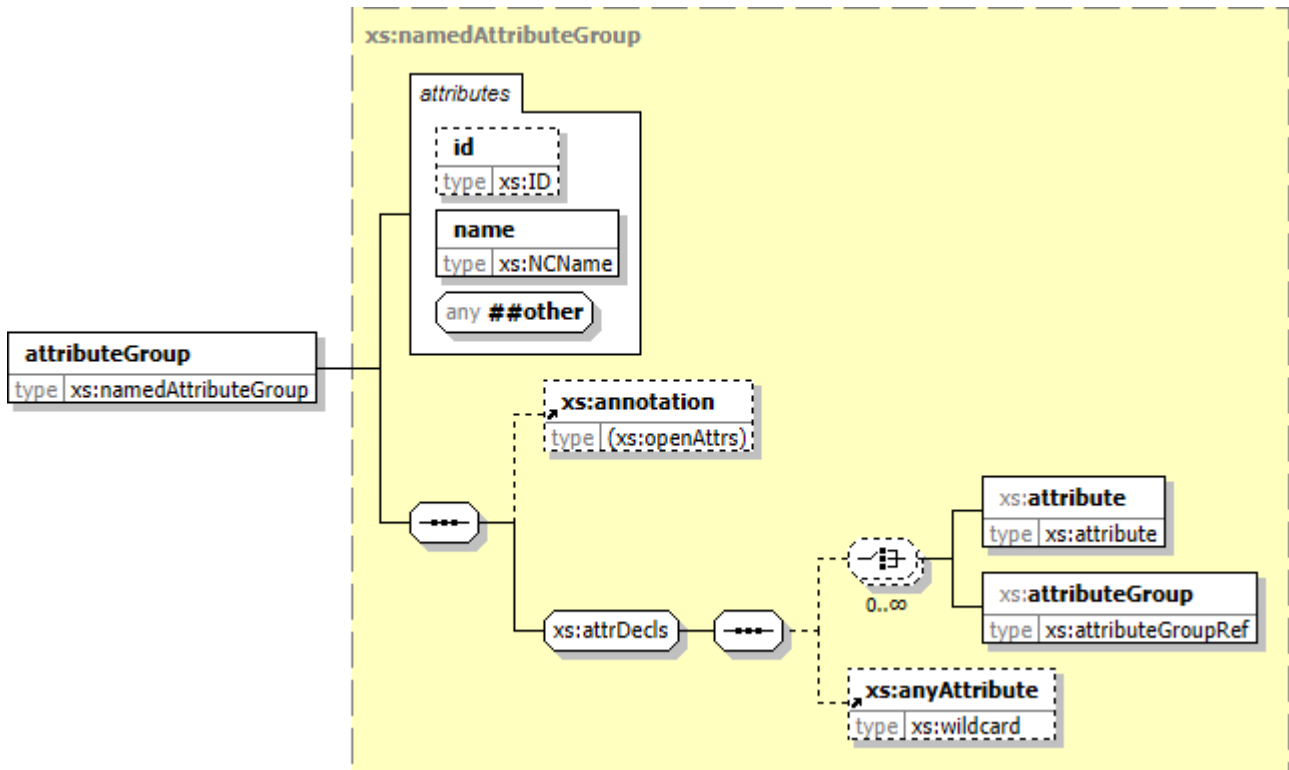
Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) [161] within [xs:attribute](#) complexType; see [XML source](#) [401]

global element

<xs:attributeGroup>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:namedAttributeGroup](#) [209]
 Content: complex, 2 attributes, attr. wildcard, 4 elements
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [33]

Component Diagram



XML Representation Summary

```
<xs:attributeGroup
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?
</xs:attributeGroup>
```

Content model elements (4):

[xs:annotation](#) [17], [xs:attribute](#) (type [xs:attribute](#)) [29],
[xs:anyAttribute](#) [23], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

Included in content model of elements (2):

[xs:redefine](#) [111], [xs:schema](#) [7]

Known Usage Locations

- Within model groups (1):

[xs:redefinable](#) [354]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-attributeGroup>

XML Source (see within schema source: p. 410)

```
<xs:element id="attributeGroup" name="attributeGroup" type="xs:namedAttributeGroup">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-attributeGroup"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [210] within [xs:namedAttributeGroup](#) complexType; see [XML source](#) [409]

{any attribute from non-schema namespace}

Defined: [locally](#) [210] within [xs:namedAttributeGroup](#) complexType; see [XML source](#) [409]

Content Element Detail (all declarations; 4/4)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [210] within [xs:namedAttributeGroup](#) complexType; see [XML source](#) [409]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

local element

`<xs:attributeGroup>` (type `xs:attributeGroupRef`)

Namespace: <http://www.w3.org/2001/XMLSchema>

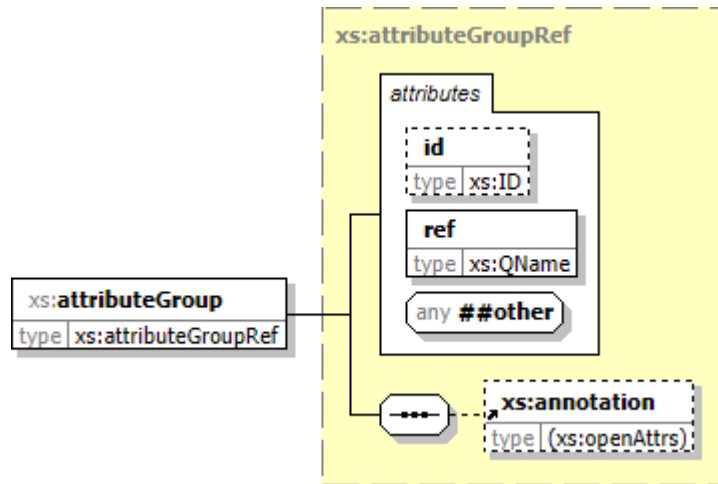
Type: `xs:attributeGroupRef` [165]

Content: complex, 2 [attributes](#), attr. [wildcard](#), 1 [element](#)

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within `xs:attrDecls` group [341] in `XMLSchema.xsd`; see [XML source](#) [34]

Component Diagram



XML Representation Summary

```
<xs:attributeGroup
  id = xs:ID
  ref = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:attributeGroup>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (7):

[xs:attributeGroup](#) [32], [xs:extension](#) (in [xs:simpleContent](#)) [70],
[xs:complexType](#) [44], [xs:restriction](#) (in [xs:complexType](#)) [118],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:restriction](#) (in [xs:simpleContent](#)) [121],
[xs:extension](#) (in [xs:complexType](#)) [67],

XML Source (see within schema source: p. 401)

```
<xs:element name="attributeGroup" type="xs:attributeGroupRef" />
```

Attribute Detail (all declarations; 3/3)

■ id

Type: `xs:ID` [295]

Use: optional

Defined: locally [155] within `xs:annotated` complexType; see [XML source](#) [397]

■ `ref`

Type: `xs:QName` [322]

Use: required

Defined: `locally` [166] within `xs:attributeGroupRef` complexType; see [XML source](#) [409]

■ *{any attribute from non-schema namespace}*

Defined: `locally` [166] within `xs:attributeGroupRef` complexType; see [XML source](#) [410]

Content Element Detail (all declarations; 1/1)

↔ `xs:annotation` [17]

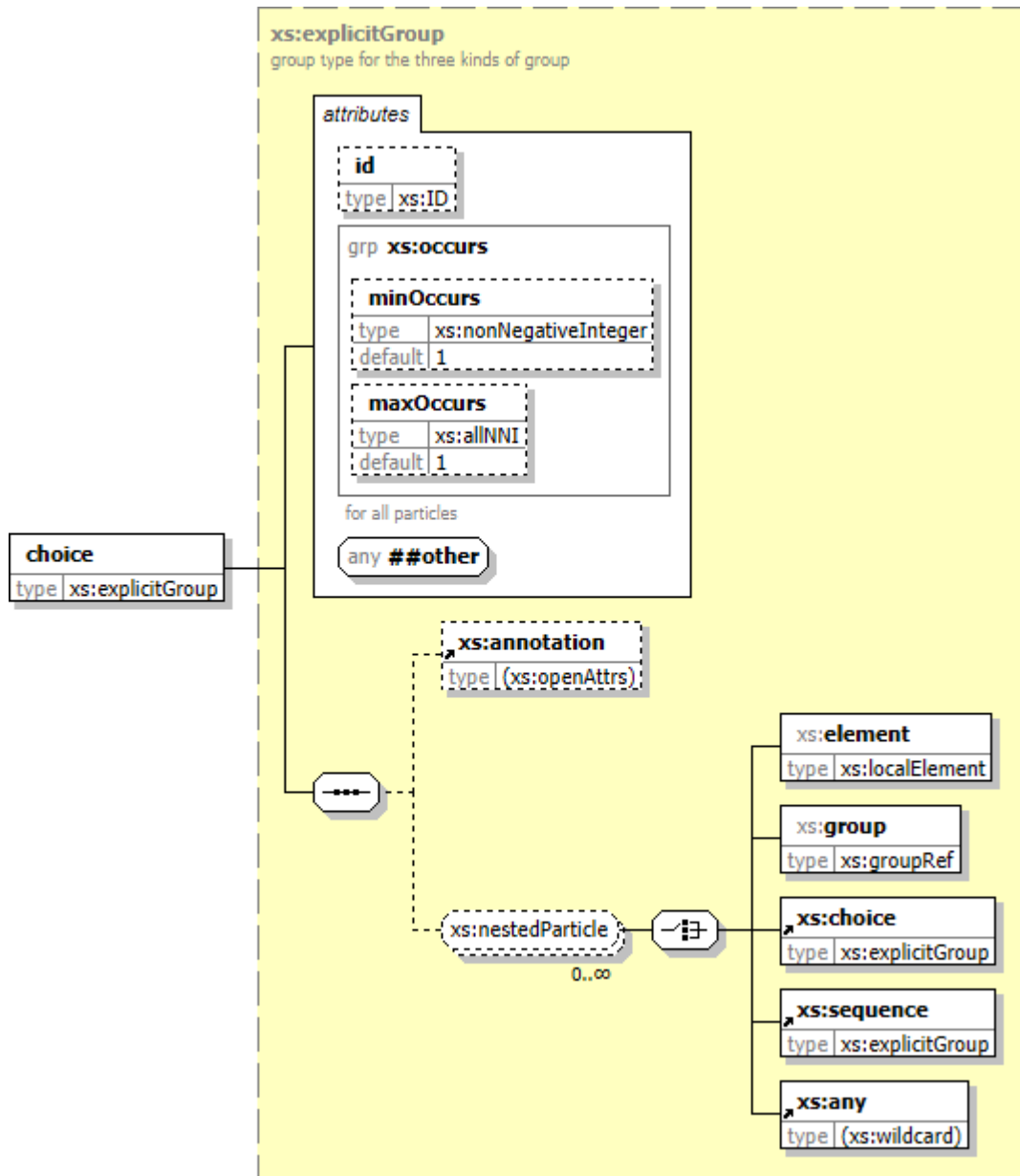
Type: `anonymous` complexType (extension of `xs:openAttrs`) [18], complex content

Defined: `by reference` [166] within `xs:attributeGroupRef` complexType; see [XML source](#) [409]

global element <xs:choice>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:explicitGroup](#) [179]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 6 [elements](#)
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [37]

Component Diagram



XML Representation Summary

```

<xs:choice
  id = xs:ID
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:element | xs:group | xs:choice | xs:sequence | xs:any)*
</xs:choice>
    
```


Content model elements (6):

[xs:annotation](#) [17], [xs:element](#) (type [xs:localElement](#)) [57],
[xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]

Included in content model of elements (8):

[xs:choice](#) [36], [xs:extension](#) (in [xs:complexContent](#)) [67],
[xs:choice](#) (in [xs:group](#)) [39], [xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:complexType](#) [44], [xs:sequence](#) [128],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:sequence](#) (in [xs:group](#)) [131]

Known Usage Locations

- **Within global complexTypes (1):**
[xs:realGroup](#) [228]
- **Within model groups (3):**
[xs:nestedParticle](#) [350], [xs:particle](#) [352], [xs:typeDefParticle](#) [362]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-choice>

XML Source (see within schema source: p. 408)

```
<xs:element id="choice" name="choice" type="xs:explicitGroup">  
  <xs:annotation>  
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-choice"/>  
  </xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 4/4)

■ id
Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs
Type: [xs:allNNI](#) [263]
Use: optional
Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

■ minOccurs
Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

■ {any attribute from non-schema namespace}

Defined: [locally](#) [181] within [xs:explicitGroup](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) [181] within [xs:explicitGroup](#) complexType; see [XML source](#) [406]

↔ [xs:any](#) [20]

Type: [anonymous](#) complexType (extension of [xs:wildcard](#)) [21], complex content

Defined: [by reference](#) [349] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

local element

`<xs:choice>` (in `xs:group`)

Namespace: <http://www.w3.org/2001/XMLSchema>

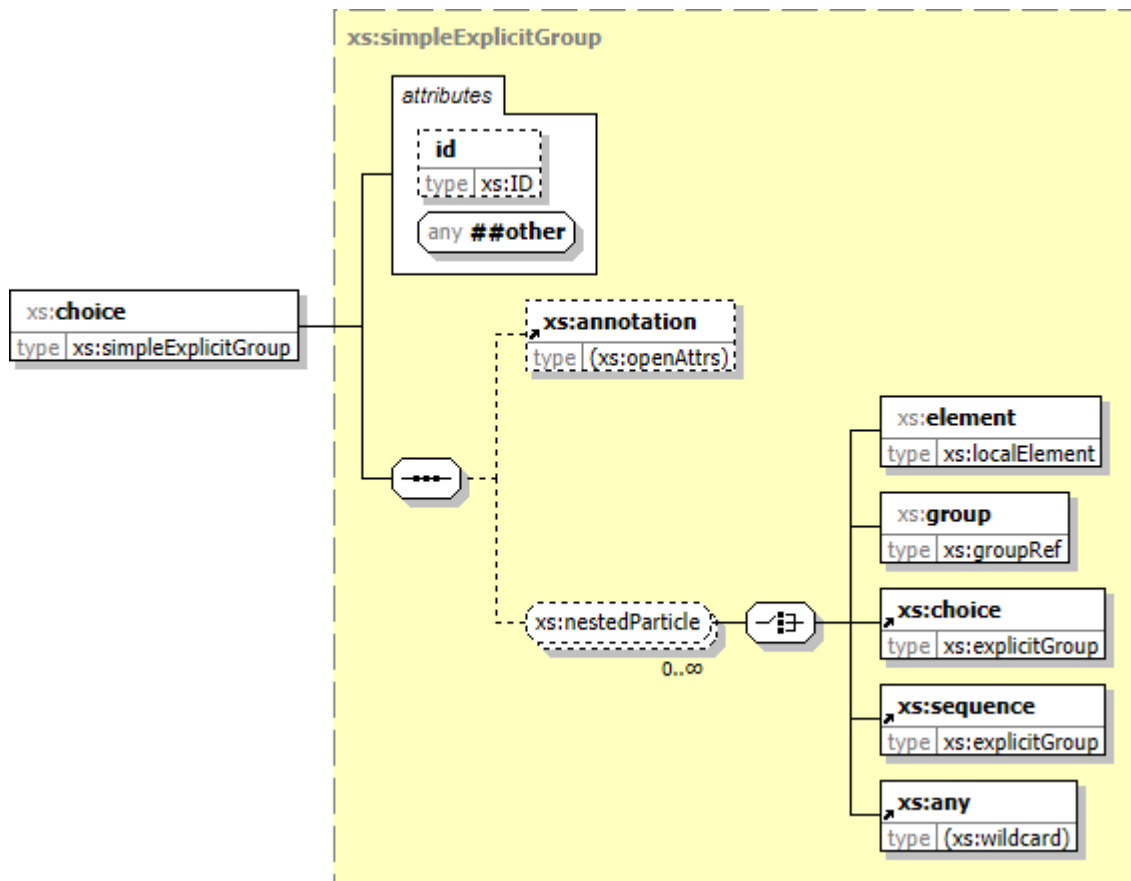
Type: `xs:simpleExplicitGroup` [233]

Content: complex, 1 attribute, attr. wildcard, 6 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within `xs:namedGroup` complexType [214] in `XMLSchema.xsd`; see [XML source](#) [39]

Component Diagram



XML Representation Summary

```
<xs:choice
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:element | xs:group | xs:choice | xs:sequence | xs:any)*
</xs:choice>
```

Content model elements (6):

`xs:annotation` [17], `xs:element` (type `xs:localElement`) [57],
`xs:any` [20], `xs:group` (type `xs:groupRef`) [79],
`xs:choice` [36], `xs:sequence` [128]

Included in content model of elements (1):

`xs:group` [77]

XML Source (see within schema source: p. 406)

```
<xs:element name="choice" type="xs:simpleExplicitGroup"/>
```

Attribute Detail (all declarations; 2/2)

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [234] within [xs:simpleExplicitGroup](#) complexType; see [XML source](#) [407]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) [234] within [xs:simpleExplicitGroup](#) complexType; see [XML source](#) [407]

↔ [xs:any](#) [20]

Type: [anonymous](#) complexType ([extension of xs:wildcard](#)) [21], complex content

Defined: [by reference](#) [349] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

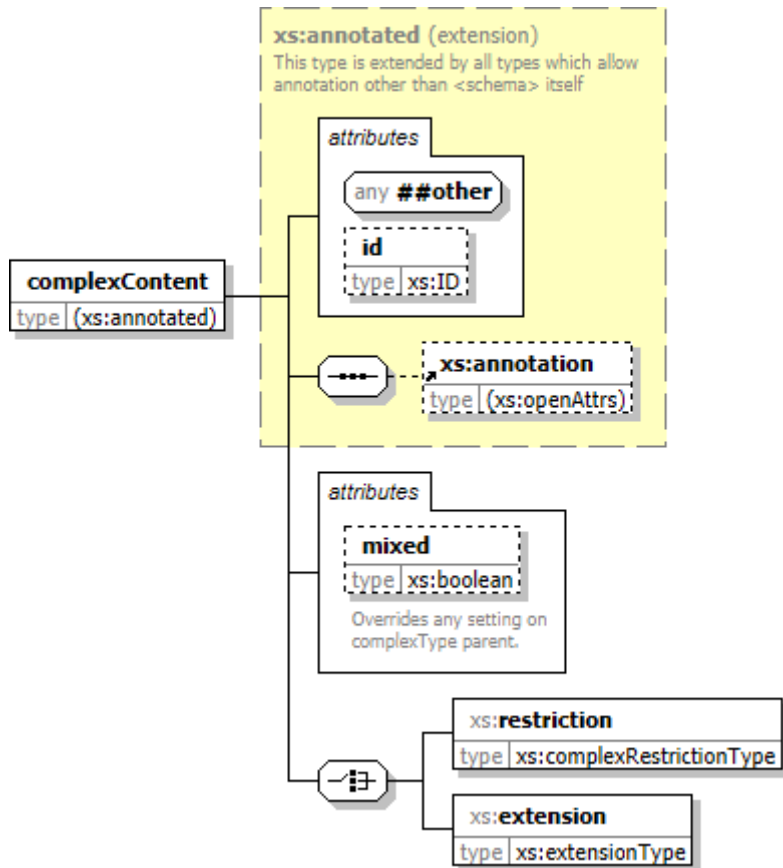
Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

global element

<xs:complexContent>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [42]
Content: complex, 2 attributes, attr. wildcard, 3 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [42]

Component Diagram



XML Representation Summary

```
<xs:complexContent
  id = xs:ID
  mixed = xs:boolean
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:restriction | xs:extension)
</xs:complexContent>
```

Content model elements (3):

[xs:annotation](#) [17], [xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:extension](#) (in [xs:complexContent](#)) [67],

Included in content model of elements (2):

[xs:complexType](#) [44], [xs:complexType](#) (type [xs:localComplexType](#)) [48]

Known Usage Locations

- Within model groups (1):
[xs:complexTypeModel](#) [343]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-complexContent>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
    
```

XML Source (see within schema source: p. 403)

```

<xs:element id="complexContent" name="complexContent">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-complexContent"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:choice>
          <xs:element name="restriction" type="xs:complexType"/>
          <xs:element name="extension" type="xs:extensionType"/>
        </xs:choice>
        <xs:attribute name="mixed" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>
              Overrides any setting on complexType parent.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

mixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: locally within (this) [xs:complexContent](#) element; see [XML source](#) [403]
 Overrides any setting on complexType parent.

{any attribute from non-schema namespace}

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 3/3)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

[xs:extension](#) [67]

Type: [xs:extensionType](#) [183], complex content
Defined: locally within ([this](#)) [xs:complexContent](#) element; see [XML source](#) [403]

[xs:restriction](#) [118]

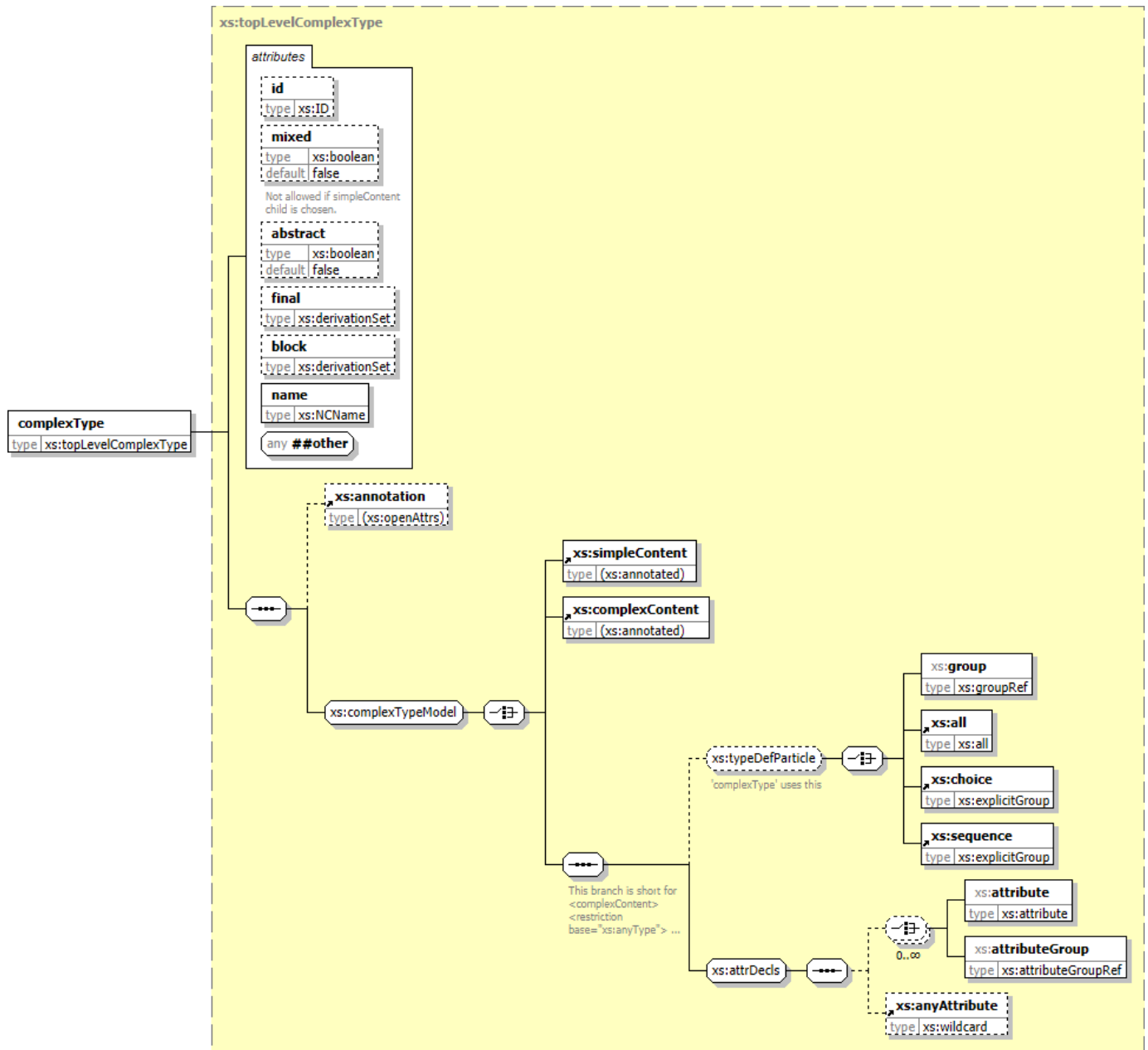
Type: [xs:complexRestrictionType](#) [167], complex content
Defined: locally within ([this](#)) [xs:complexContent](#) element; see [XML source](#) [403]

global element

<xs:complexType>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:topLevelComplexType](#) [249]
Content: complex, 6 attributes, attr. wildcard, 10 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [45]

Component Diagram



XML Representation Summary

```
<xs:complexType
  id = xs:ID
  mixed = xs:boolean : "false"
  abstract = xs:boolean : "false"
  final = ("#all" | list of ("extension" | "restriction"))
  block = ("#all" | list of ("extension" | "restriction"))
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleContent | xs:complexContent | ((xs:group | xs:all | xs:choice |
  xs:sequence)?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?))
</xs:complexType>
```

Content model elements (10):

- xs:all [12],
- xs:annotation [17],
- xs:anyAttribute [23],
- xs:attribute (type xs:attribute) [29],
- xs:attributeGroup (type xs:attributeGroupRef) [34],
- xs:choice [36],
- xs:complexContent [41],
- xs:group (type xs:groupRef) [79],
- xs:sequence [128],
- xs:simpleContent [133]

Included in content model of elements (2):

- xs:redefine [111], xs:schema [7]

Known Usage Locations

- Within model groups (1):

- xs:redefinable [354]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-complexType>

XML Source (see within schema source: p. 404)

```
<xs:element id="complexType" name="complexType" type="xs:topLevelComplexType">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-complexType"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 7/7)

abstract

Type: xs:boolean [270]
Use: optional
Defined: locally [172] within xs:complexType complexType; see XML source [402]

Attribute Value

Default: "false"

block

Type: xs:derivationSet [279]
Use: optional
Defined: locally [172] within xs:complexType complexType; see XML source [402]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

final

Type: [xs:derivationSet](#) [279]
Use: optional
Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

mixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Not allowed if simpleContent child is chosen.
 May be overridden by setting on complexContent child.

Attribute Value

Default: "false"

name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [251] within [xs:topLevelComplexType](#) complexType; see [XML source](#) [402]

{any attribute from non-schema namespace}

Defined: [locally](#) [251] within [xs:topLevelComplexType](#) complexType; see [XML source](#) [402]

Content Element Detail (all declarations; 10/10)

[xs:all](#) [12]

Type: [xs:all](#) [150], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [251] within [xs:topLevelComplexType](#) complexType; see [XML source](#) [402]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:complexContent](#) [41]

Type: [anonymous complexType](#) ([extension of \[xs:annotated\]\(#\)](#)) [42], complex content

Defined: [by reference](#) [343] within [xs:complexTypeModel](#) group; see [XML source](#) [401]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:simpleContent](#) [133]

Type: [anonymous complexType](#) ([extension of \[xs:annotated\]\(#\)](#)) [134], complex content

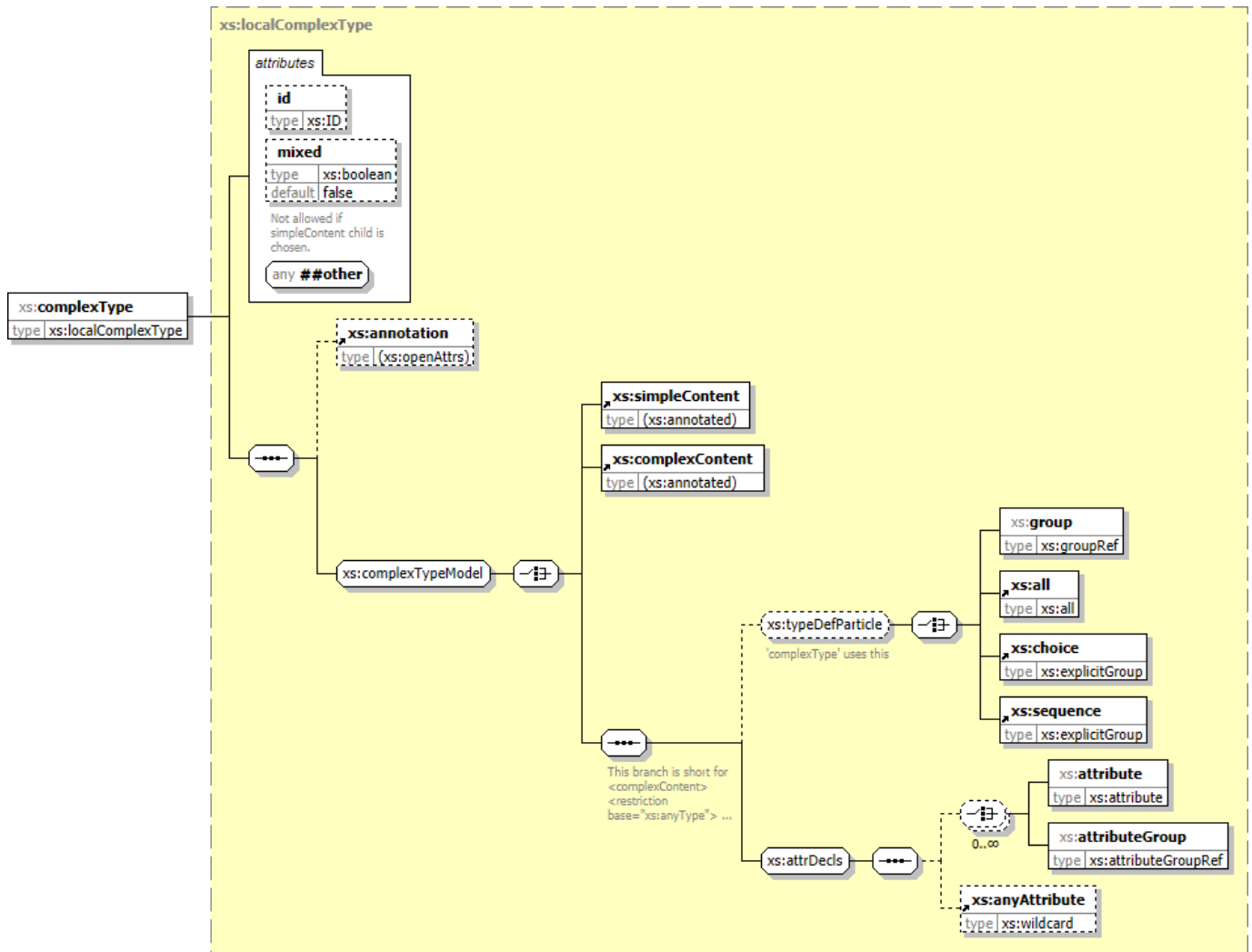
Defined: [by reference](#) [343] within [xs:complexTypeModel](#) group; see [XML source](#) [401]

unified local element

`<xs:complexType>` (type `xs:localComplexType`)

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: `xs:localComplexType` [197]
Content: complex, 2 attributes, attr. wildcard, 10 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: locally at 4 locations in `XMLSchema.xsd`

Component Diagram



XML Representation Summary

```
<xs:complexType
  id = xs:ID
  mixed = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:simpleContent | xs:complexContent | ((xs:group | xs:all | xs:choice |
    xs:sequence)?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?))
</xs:complexType>
```

Content model elements (10):

- [xs:all](#) [12],
- [xs:annotation](#) [17],
- [xs:anyAttribute](#) [23],
- [xs:attribute](#) (type `xs:attribute`) [29],
- [xs:choice](#) [36],
- [xs:complexContent](#) [41],
- [xs:group](#) (type `xs:groupRef`) [79],
- [xs:sequence](#) [128],

[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34], [xs:simpleContent](#) [133]

Included in content model of elements (3):

[xs:element](#) [53], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]
[xs:element](#) (type [xs:localElement](#)) [57],

Definition Locations

- **Within global complexTypes (4):**

[xs:element](#) [178], [xs:localElement](#) [204], [xs:narrowMaxMin](#) [219], [xs:topLevelElement](#) [256]

Annotations (1) (by all definition locations)

Locations (4):

within [xs:element](#) complexType [178], within [xs:topLevelElement](#) complexType [256], within [xs:localElement](#) complexType [204], within [xs:narrowMaxMin](#) complexType [219]

Annotation:

Attribute Detail (all declarations; 3/3)

■ [id](#)

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ [mixed](#)

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Not allowed if simpleContent child is chosen.
May be overridden by setting on complexContent child.

Attribute Value

Default: "false"

■ [{any attribute from non-schema namespace}](#)

Defined: [locally](#) [199] within [xs:localComplexType](#) complexType; see [XML source](#) [402]

Content Element Detail (all declarations; 10/10)

↔ [xs:all](#) [12]

Type: [xs:all](#) [150], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [199] within [xs:localComplexType](#) complexType; see [XML source](#) [402]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ `xs:attribute` [29]

Type: `xs:attribute` [158], complex content
Defined: `locally` [340] within `xs:attrDecls` group; see [XML source](#) [401]

↔ `xs:attributeGroup` [34]

Type: `xs:attributeGroupRef` [165], complex content
Defined: `locally` [341] within `xs:attrDecls` group; see [XML source](#) [401]

↔ `xs:choice` [36]

Type: `xs:explicitGroup` [179], complex content
Defined: `by reference` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:complexContent` [41]

Type: `anonymous complexType (extension of xs:annotated)` [42], complex content
Defined: `by reference` [343] within `xs:complexTypeModel` group; see [XML source](#) [401]

↔ `xs:group` [79]

Type: `xs:groupRef` [192], complex content
Defined: `locally` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:sequence` [128]

Type: `xs:explicitGroup` [179], complex content
Defined: `by reference` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:simpleContent` [133]

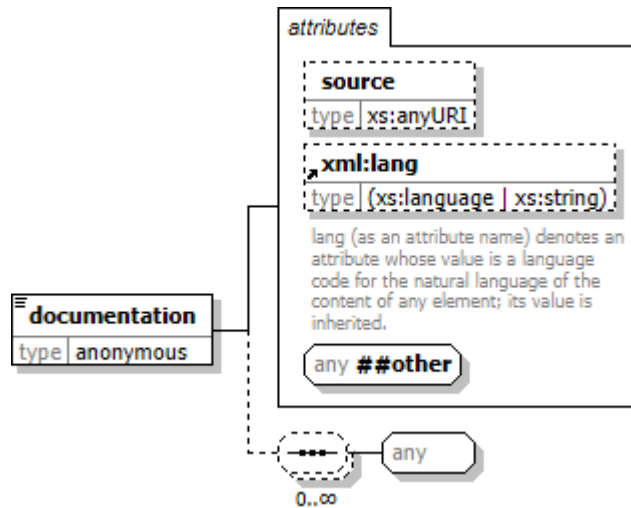
Type: `anonymous complexType (extension of xs:annotated)` [134], complex content
Defined: `by reference` [343] within `xs:complexTypeModel` group; see [XML source](#) [401]

global element

<xs:documentation>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [anonymous](#) complexType [412]
Content: mixed (*allows character data*), 2 [attributes](#), attr. [wildcard](#), elem. [wildcard](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [51]

Component Diagram



XML Representation Summary

```
<xs:documentation
  source = xs:anyURI
  xml:lang = (xs:language | "")
  {any attribute from non-schema namespace}
>
Content: {text} × {any}*
</xs:documentation>
```

Included in content model of elements (1):

[xs:annotation](#) [17]

Known Usage Locations

- Within anonymous complexTypes of elements (1):

[xs:annotation](#) [19]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-documentation>

XML Source (see within schema source: p. 412)

```
<xs:element id="documentation" name="documentation">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-documentation"/>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:sequence maxOccurs="unbounded" minOccurs="0">
      <xs:any processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="source" type="xs:anyURI"/>
  </xs:complexType>
</xs:element>
```

```
<xs:attribute ref="xml:lang"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
</xs:element>
```

Attribute Detail (all declarations; 3/3)

■ source

Type: [xs:anyURI](#) [265]
Use: optional
Defined: locally within [\(this\) xs:documentation](#) element; see [XML source](#) [412]

■ xml:lang [370]

Type: [anonymous simpleType](#) ([union of \(xs:language | restriction of xs:string\)](#)) [370]
Use: optional
Defined: by reference within [\(this\) xs:documentation](#) element; see [XML source](#) [412]

Attribute Value

```
xs:language | ""
```

■ {any attribute from non-schema namespace}

Defined: locally within [\(this\) xs:documentation](#) element; see [XML source](#) [412]

Content Element Detail (all declarations; 1/1)

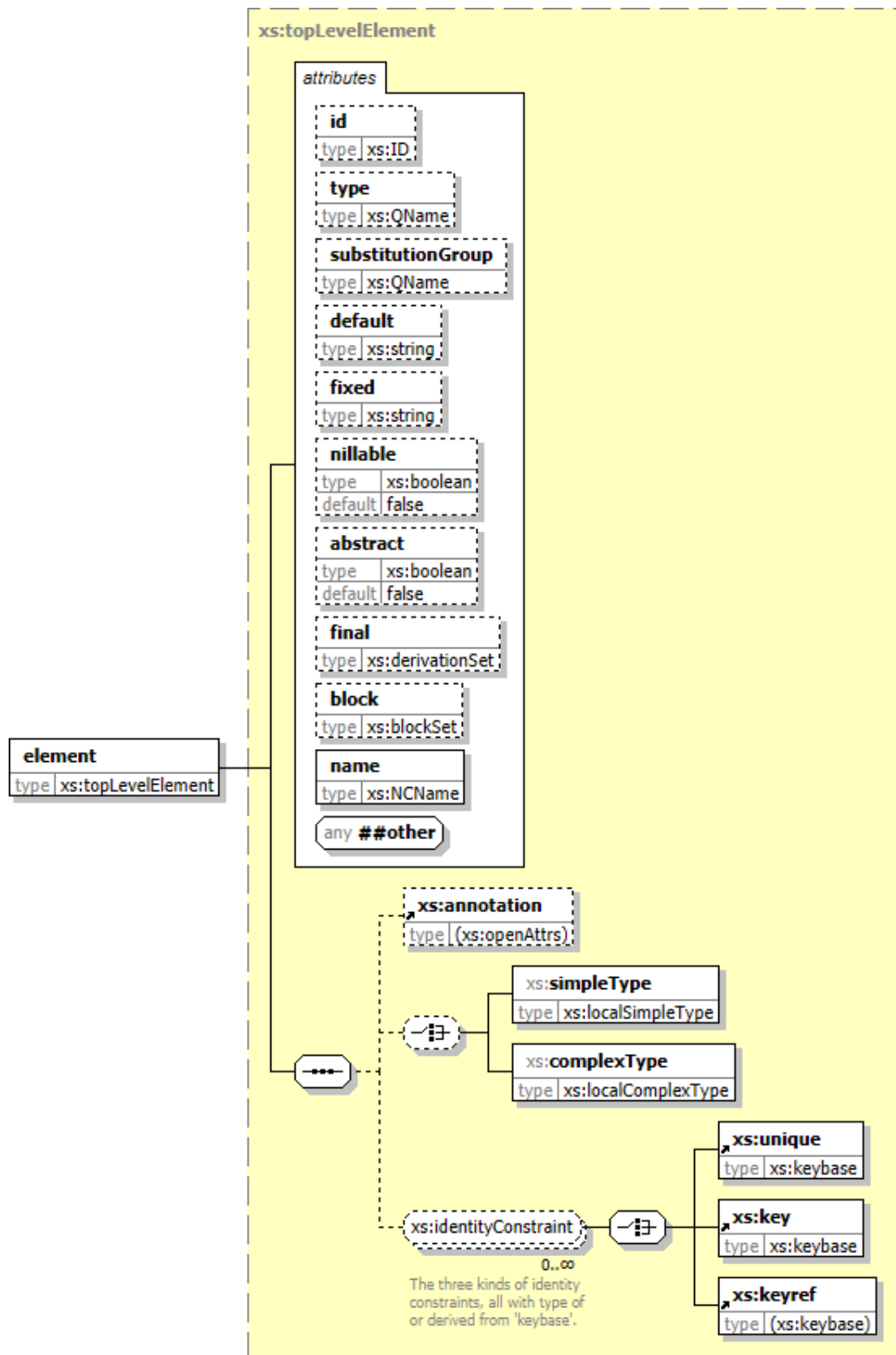
↔ {any element from any namespace}

Defined: locally within [\(this\) xs:documentation](#) element; see [XML source](#) [412]

global element <xs:element>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:topLevelElement](#) [253]
Content: complex, 10 [attributes](#), attr. [wildcard](#), 6 [elements](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [54]

Component Diagram



XML Representation Summary

```
<xs:element
  id                = xs:ID
  type              = xs:QName
  substitutionGroup = xs:QName
  default           = xs:string
  fixed             = xs:string
  nillable          = xs:boolean : "false"
  abstract          = xs:boolean : "false"
  final             = ("#all" | list of ("extension" | "restriction"))
  block             = ("#all" | list of ("extension" | "restriction" | "substitution"))
  name              = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</xs:element>
```

Content model elements (6):

[xs:annotation](#) [17], [xs:keyref](#) [87],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
[xs:key](#) [85], [xs:unique](#) [145]

Included in content model of elements (1):

[xs:schema](#) [7]

Known Usage Locations

- Within model groups (1):

[xs:schemaTop](#) [356]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-element>

XML Source (see within schema source: p. 405)

```
<xs:element id="element" name="element" type="xs:topLevelElement">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-element"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 11/11)

abstract

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

block

Type: [xs:blockSet](#) [268]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "substitution")
```

■ default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

■ final

Type: [xs:derivationSet](#) [279]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

■ fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [255] within [xs:topLevelElement](#) complexType; see [XML source](#) [405]

■ nillable

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

■ substitutionGroup

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [256] within [xs:topLevelElement](#) complexType; see [XML source](#) [405]

Content Element Detail (all declarations; 6/6)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [256] within [xs:topLevelElement](#) complexType; see [XML source](#) [405]

[xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content
Defined: [locally](#) [256] within [xs:topLevelElement](#) complexType; see [XML source](#) [405]

[xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content
Defined: [by reference](#) [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

[xs:keyref](#) [87]

Type: [anonymous](#) complexType ([extension of xs:keybase](#)) [88], complex content
Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

[xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) [256] within [xs:topLevelElement](#) complexType; see [XML source](#) [405]

[xs:unique](#) [145]

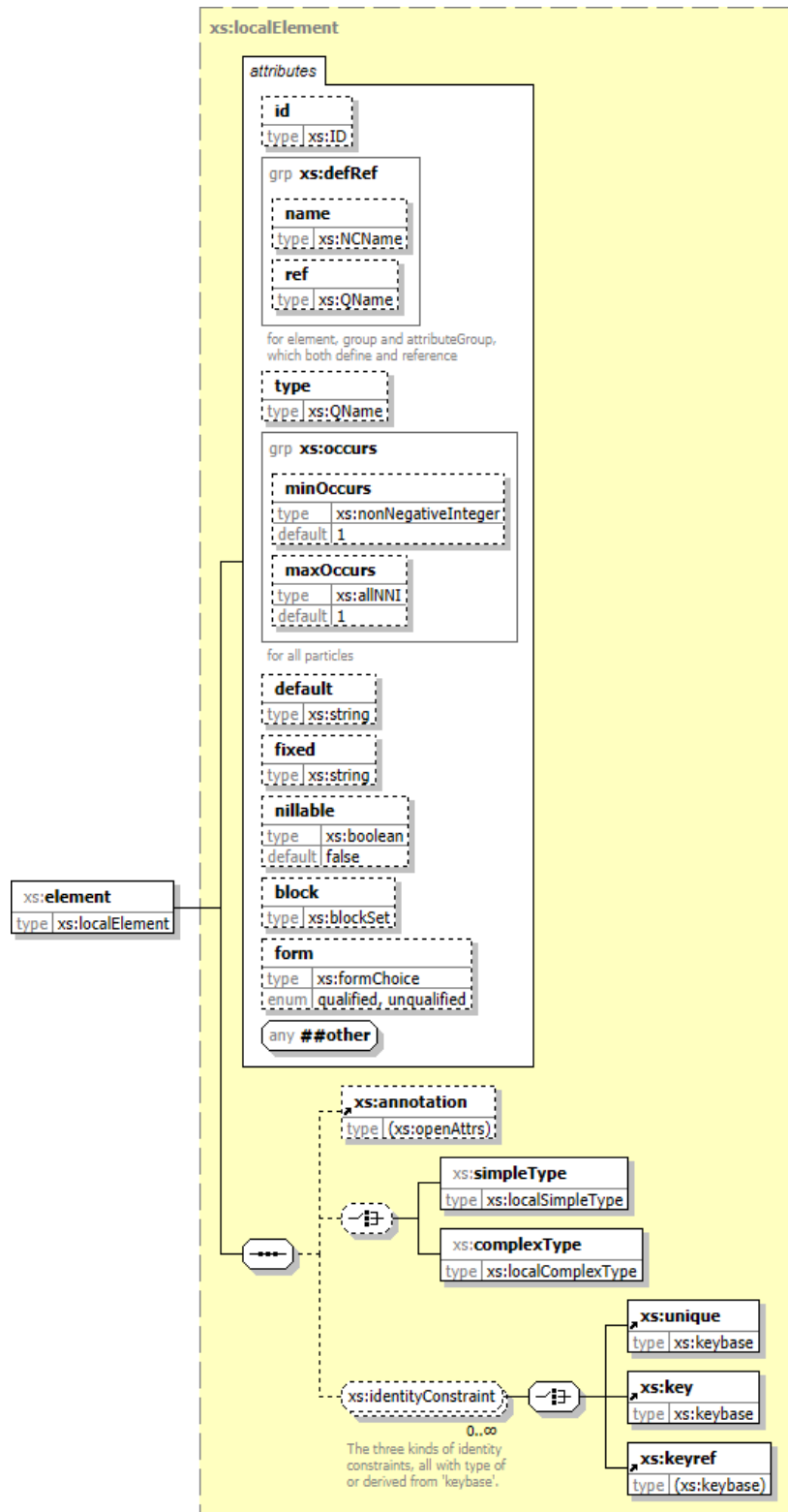
Type: [xs:keybase](#) [195], complex content
Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

unified local element

<xs:element> (type xs:localElement)

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:localElement](#) [201]
Content: complex, 11 [attributes](#), attr. [wildcard](#), 6 [elements](#)
Block: "#all" (blocks all substitutions of this element or its type)
Defined: locally at 2 [locations](#) in [XMLSchema.xsd](#)

Component Diagram



XML Representation Summary

```
<xs:element
  id           = xs:ID
  name        = xs:NCName
  ref         = xs:QName
  type        = xs:QName
  minOccurs   = xs:nonNegativeInteger : "1"
  maxOccurs   = (xs:nonNegativeInteger | "unbounded") : "1"
  default     = xs:string
  fixed       = xs:string
  nillable    = xs:boolean : "false"
  block       = ("#all" | list of ("extension" | "restriction" | "substitution"))
  form        = ("qualified" | "unqualified")
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</xs:element>
```

Content model elements (6):

- [xs:annotation](#) [17], [xs:keyref](#) [87],
- [xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
- [xs:key](#) [85], [xs:unique](#) [145]

Included in content model of elements (4):

- [xs:choice](#) [36], [xs:sequence](#) [128],
- [xs:choice](#) (in [xs:group](#)) [39], [xs:sequence](#) (in [xs:group](#)) [131]

Definition Locations

- Within model groups (2):

- [xs:nestedParticle](#) [350], [xs:particle](#) [352]

Annotations (1) (by all definition locations)

Locations (2):

- within [xs:nestedParticle](#) group [350], within [xs:particle](#) group [352]

Annotation:

Attribute Detail (all declarations; 12/12)

block

Type: [xs:blockSet](#) [268]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "substitution")
```

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

■ form

Type: [xs:formChoice](#) [286]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

`enumeration of xs:NMTOKEN`

Enumeration: "qualified", "unqualified"

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs

Type: [xs:allNNI](#) [263]
Use: optional
Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

`xs:nonNegativeInteger | "unbounded"`

Default: "1"

■ minOccurs

Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

■ name

Type: [xs:NCName](#) [308]
Use: optional
Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ nillable

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

■ ref

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [204] within [xs:localElement](#) complexType; see [XML source](#) [405]

Content Element Detail (all declarations; 6/6)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [204] within [xs:localElement](#) complexType; see [XML source](#) [405]

[xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content
Defined: [locally](#) [204] within [xs:localElement](#) complexType; see [XML source](#) [405]

[xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content
Defined: [by reference](#) [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

[xs:keyref](#) [87]

Type: [anonymous](#) complexType (extension of [xs:keybase](#)) [88], complex content
Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

[xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) [205] within [xs:localElement](#) complexType; see [XML source](#) [405]

[xs:unique](#) [145]

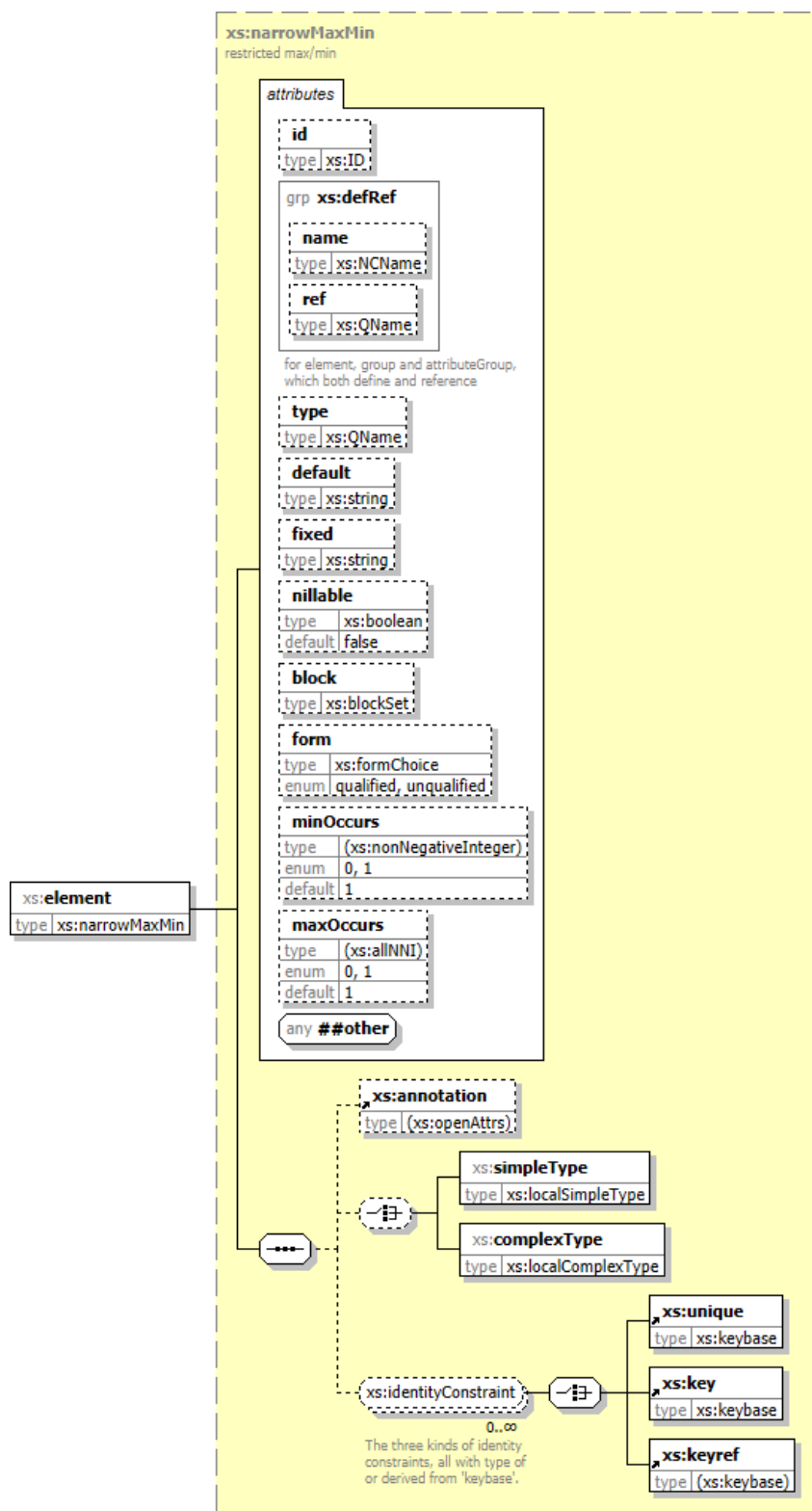
Type: [xs:keybase](#) [195], complex content
Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

local element

<xs:element> (type xs:narrowMaxMin)

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:narrowMaxMin](#) [215]
Content: complex, 11 [attributes](#), attr. [wildcard](#), 6 [elements](#)
Block: "#all" (blocks all substitutions of this element or its type)
Defined: locally within [xs:allModel](#) group [339] in [XMLSchema.xsd](#); see [XML source](#) [62]

Component Diagram



XML Representation Summary

```
<xs:element
  id           = xs:ID
  name        = xs:NCName
  ref         = xs:QName
  type        = xs:QName
  default     = xs:string
  fixed       = xs:string
  nillable    = xs:boolean : "false"
  block      = ("#all" | list of ("extension" | "restriction" | "substitution"))
  form       = ("qualified" | "unqualified")
  minOccurs  = ("0" | "1") : "1"
  maxOccurs  = ("0" | "1") : "1"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</xs:element>
```

Content model elements (6):

- [xs:annotation](#) [17],
- [xs:complexType](#) (type [xs:localComplexType](#)) [48],
- [xs:key](#) [85],
- [xs:keyref](#) [87],
- [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
- [xs:unique](#) [145]

Included in content model of elements (2):

- [xs:all](#) [12],
- [xs:all](#) (in [xs:group](#)) [15]

XML Source (see within schema source: p. 407)

```
<xs:element name="element" type="xs:narrowMaxMin"/>
```

Attribute Detail (all declarations; 12/12)

block

Type: [xs:blockSet](#) [268]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "substitution")
```

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

form

Type: [xs:formChoice](#) [286]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs

Type: [anonymous simpleType](#) (restriction of [xs:allNNI](#)) [63]
Use: optional
Defined: [locally](#) [217] within [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

Attribute Value

enumeration of ([xs:nonNegativeInteger](#) | "unbounded")

Enumeration: "0", "1"
Default: "1"

Anonymous simpleType

Type Derivation Tree
union of ([xs:nonNegativeInteger](#) | restriction of [xs:NMTOKEN](#))
└─ [xs:allNNI](#) [263] (restriction)
 └─ [simpleType](#) [218]

■ minOccurs

Type: [anonymous simpleType](#) (restriction of [xs:nonNegativeInteger](#)) [63]
Use: optional
Defined: [locally](#) [218] within [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

Attribute Value

enumeration of [xs:nonNegativeInteger](#)

Enumeration: "0", "1"
Default: "1"

Anonymous simpleType

Type Derivation Tree
[xs:anySimpleType](#) (restriction)
└─ [xs:decimal](#) [275] (restriction)
 └─ [xs:integer](#) [299] (restriction)
 └─ [xs:nonNegativeInteger](#) [314] (restriction)
 └─ [simpleType](#) [218]

■ name

Type: [xs:NCName](#) [308]
Use: optional
Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ nillable

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

■ ref

Type: [xs:QName](#) [322]
Use: optional

Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ type

Type: [xs:QName](#) [322]

Use: optional

Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [218] within [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) [219] within [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

↔ [xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content

Defined: [locally](#) [219] within [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

↔ [xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content

Defined: [by reference](#) [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:keyref](#) [87]

Type: [anonymous](#) complexType (extension of [xs:keybase](#)) [88], complex content

Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content

Defined: [locally](#) [219] within [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

↔ [xs:unique](#) [145]

Type: [xs:keybase](#) [195], complex content

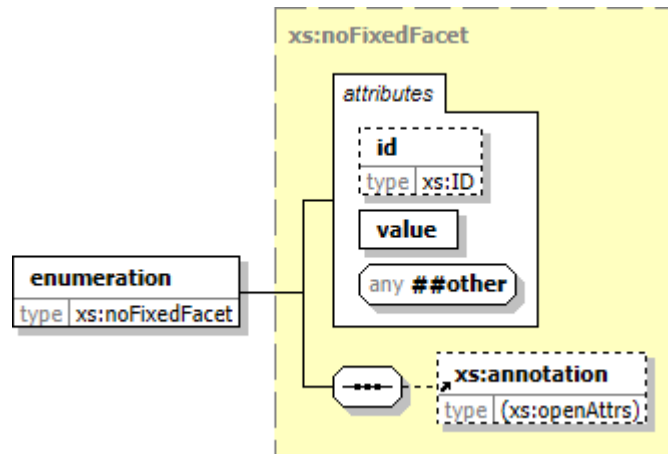
Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

global element

<xs:enumeration>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:noFixedFacet](#) [220]
 Content: complex, 2 [attributes](#), attr. [wildcard](#), 1 [element](#)
 Block: "#all" (*blocks all substitutions of this element or its type*)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [65]

Component Diagram



XML Representation Summary

```
<xs:enumeration
  id = xs:ID
  value = xs:anySimpleType
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:enumeration>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [345]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-enumeration>

XML Source (see within schema source: p. 426)

```
<xs:element id="enumeration" name="enumeration" type="xs:noFixedFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-enumeration"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 3/3)

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ value

Type: [xs:anySimpleType](#)

Use: required

Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

■ *{any attribute from non-schema namespace}*

Defined: [locally](#) [221] within [xs:noFixedFacet](#) complexType; see [XML source](#) [425]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of \[xs:openAttrs\]\(#\)](#)) [18], complex content

Defined: [by reference](#) [221] within [xs:noFixedFacet](#) complexType; see [XML source](#) [425]

local element

`<xs:extension>` (in `xs:complexContent`)

Namespace: <http://www.w3.org/2001/XMLSchema>

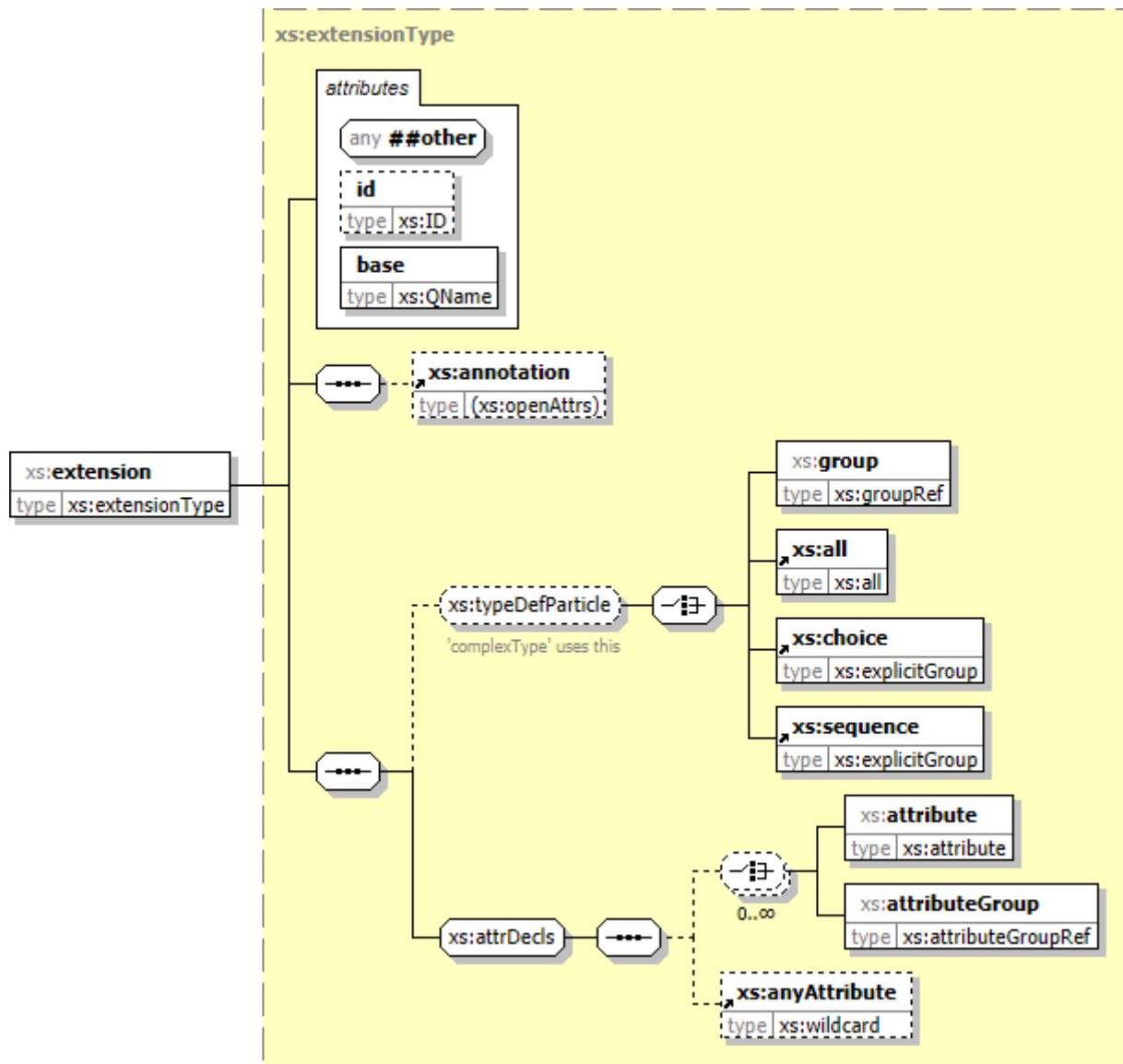
Type: `xs:extensionType` [183]

Content: complex, 2 attributes, attr. wildcard, 8 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within `xs:complexContent` element [43] in `XMLSchema.xsd`; see [XML source](#) [68]

Component Diagram



XML Representation Summary

```

<xs:extension
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:group | xs:all | xs:choice | xs:sequence)?, (xs:attribute |
    xs:attributeGroup)*, xs:anyAttribute?
</xs:extension>
    
```

Content model elements (8):

[xs:all](#) [12], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34],
[xs:annotation](#) [17], [xs:choice](#) [36],
[xs:anyAttribute](#) [23], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:attribute](#) (type [xs:attribute](#)) [29], [xs:sequence](#) [128]

Included in content model of elements (1):

[xs:complexContent](#) [41]

XML Source (see within schema source: p. 403)

```
<xs:element name="extension" type="xs:extensionType"/>
```

Attribute Detail (all declarations; 3/3)

base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [184] within [xs:extensionType](#) complexType; see [XML source](#) [403]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 8/8)

[xs:all](#) [12]

Type: [xs:all](#) [150], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

local element

`<xs:extension>` (in `xs:simpleContent`)

Namespace: <http://www.w3.org/2001/XMLSchema>

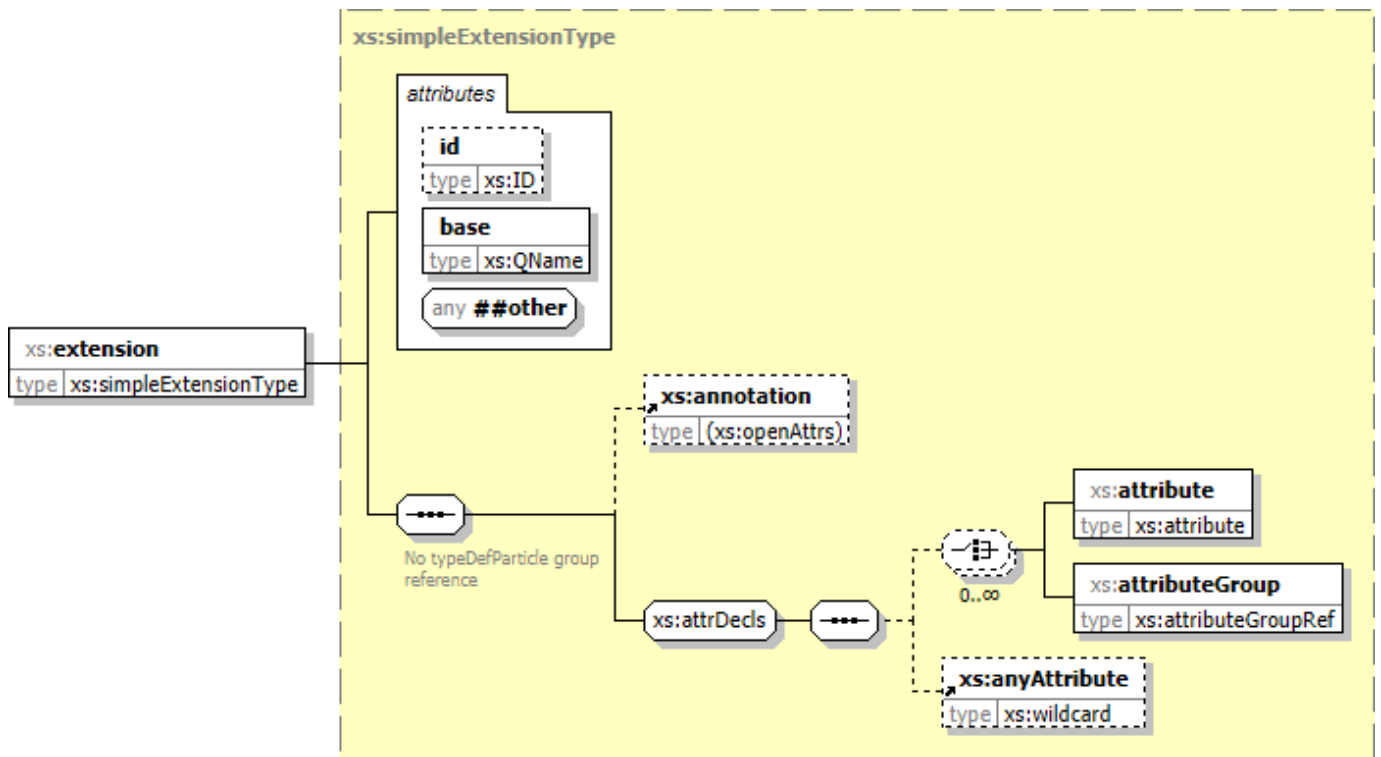
Type: `xs:simpleExtensionType` [236]

Content: complex, 2 attributes, attr. wildcard, 4 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within `xs:simpleContent` element [134] in `XMLSchema.xsd`; see [XML source](#) [70]

Component Diagram



XML Representation Summary

```
<xs:extension
  id       = xs:ID
  base    = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?
</xs:extension>
```

Content model elements (4):

`xs:annotation` [17], `xs:attribute` (type `xs:attribute`) [29],
`xs:anyAttribute` [23], `xs:attributeGroup` (type `xs:attributeGroupRef`) [34]

Included in content model of elements (1):

`xs:simpleContent` [133]

XML Source (see within schema source: p. 404)

```
<xs:element name="extension" type="xs:simpleExtensionType"/>
```

Attribute Detail (all declarations; 3/3)

base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [184] within [xs:extensionType](#) complexType; see [XML source](#) [403]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: [locally](#) [237] within [xs:simpleExtensionType](#) complexType; see [XML source](#) [404]

Content Element Detail (all declarations; 4/4)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of `xs:openAttrs`](#)) [18], complex content
Defined: [by reference](#) [237] within [xs:simpleExtensionType](#) complexType; see [XML source](#) [404]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

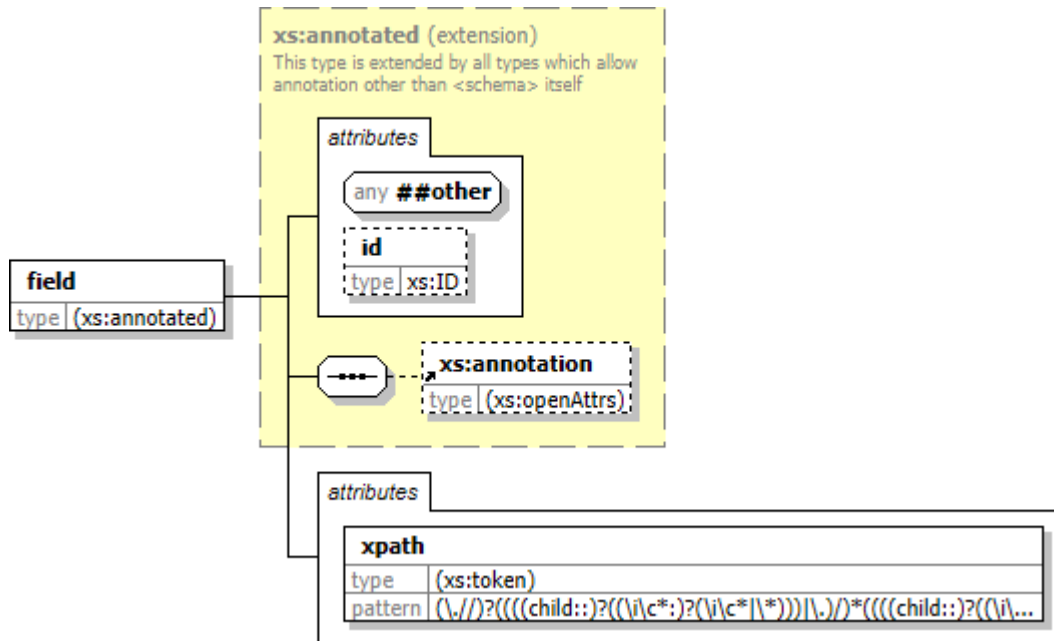
↔ [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

global element <xs:field>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [73]
Content: complex, 2 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [73]

Component Diagram



XML Representation Summary

```
<xs:field
  id = xs:ID
  xpath = xs:token
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:field>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (3):

[xs:key](#) [85], [xs:keyref](#) [87], [xs:unique](#) [145]

Known Usage Locations

- Within global complexTypes (1):

[xs:keybase](#) [196]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-field>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
  
```

XML Source (see within schema source: p. 411)

```

<xs:element id="field" name="field">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-field"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="xpath" use="required">
          <xs:simpleType>
            <xs:annotation>
              <xs:documentation>
                A subset of XPath expressions for use
                in fields
              </xs:documentation>
            </xs:annotation>
            <xs:documentation>
              A utility type, not for public
              use
            </xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:token">
            <xs:annotation>
              <xs:documentation>
                The following pattern is intended to allow XPath
                expressions per the same EBNF as for selector,
                with the following change:
                Path ::= ('./')? ( Step '/' )* ( Step | '@' NameTest )
              </xs:documentation>
            </xs:annotation>
            <xs:pattern
              value="(\./)?((((child:)?((\i\c*?)?(\i\c*|\*))|\.)/*((((child:)?((\i\c*?)?(\i\c*|\*))|\.)|((attribute::|@)((\i\c*?)?(\i\c*|\*)))|(\./)?((((child:)?((\i\c*?)?(\i\c*|\*))|\.)/*((((child:)?((\i\c*?)?(\i\c*|\*))|\.)|((attribute::|@)((\i\c*?)?(\i\c*|\*)))))))*"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
  
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

xpath

Type: [anonymous](#) simpleType ([restriction of xs:token](#)) [74]
Use: required
Defined: locally within ([this](#)) [xs:field](#) element; see [XML source](#) [411]

Attribute Value

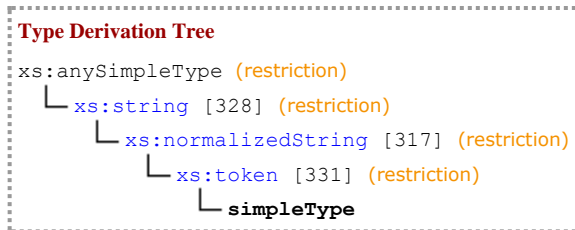
```

<xs:token
  
```

Pattern: (\./)?((((child:)?((\i\c*?)?(\i\c*|*))|\.)/*((((child:)?((\i\c*?)?(\i\c*|*))|\.)|((attribute::|@)((\i\c*?)?(\i\c*|*)))|(\./)?((((child:)?((\i\c*?)?(\i\c*|*))|\.)|((attribute::|@)((\i\c*?)?(\i\c*|*)))))))*

```
attribute::|@)((\i\c*:?(\i\c*|\*)))(\|(\./|/)?(((child::)?(\i\c*:?(\i\c*|\*))|\./)))*(((child::)?(\i\c*:?(\i\c*|\*))|\./)|((attribute::|@)((\i\c*:?(\i\c*|\*)))))*)
```

Anonymous simpleType



Annotation 1 [src, p. 411]:

A subset of XPath expressions for use in fields

Annotation 2 [src, p. 411]:

A utility type, not for public use

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content

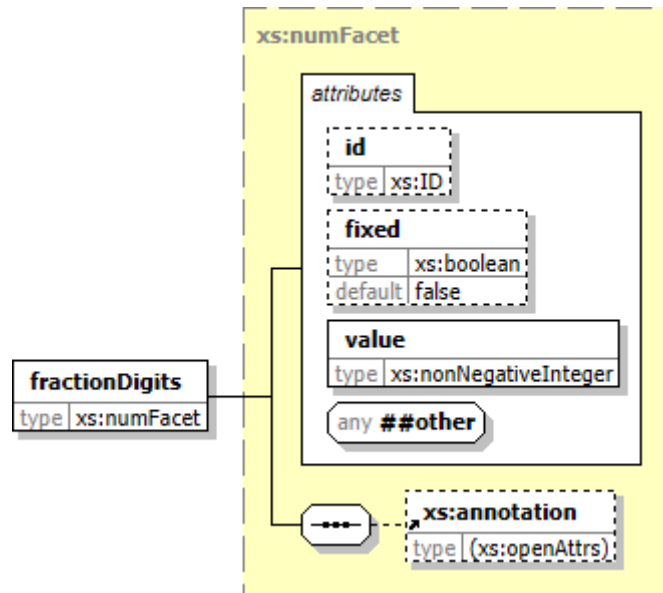
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:fractionDigits>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: `xs:numFacet` [222]
Content: complex, 3 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in `XMLSchema.xsd`; see [XML source](#) [76]

Component Diagram



XML Representation Summary

```
<xs:fractionDigits
  id = xs:ID
  fixed = xs:boolean : "false"
  value = xs:nonNegativeInteger
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:fractionDigits>
```

Content model elements (1):

`xs:annotation` [17]

Included in content model of elements (2):

`xs:restriction` [114], `xs:restriction` (in `xs:simpleContent`) [121]

Known Usage Locations

- Within model groups (1):

`xs:facets` [345]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-fractionDigits>

XML Source (see within schema source: p. 426)

```
<xs:element id="fractionDigits" name="fractionDigits" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-fractionDigits"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:nonNegativeInteger](#) [314]
Use: required
Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

{any attribute from non-schema namespace}

Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

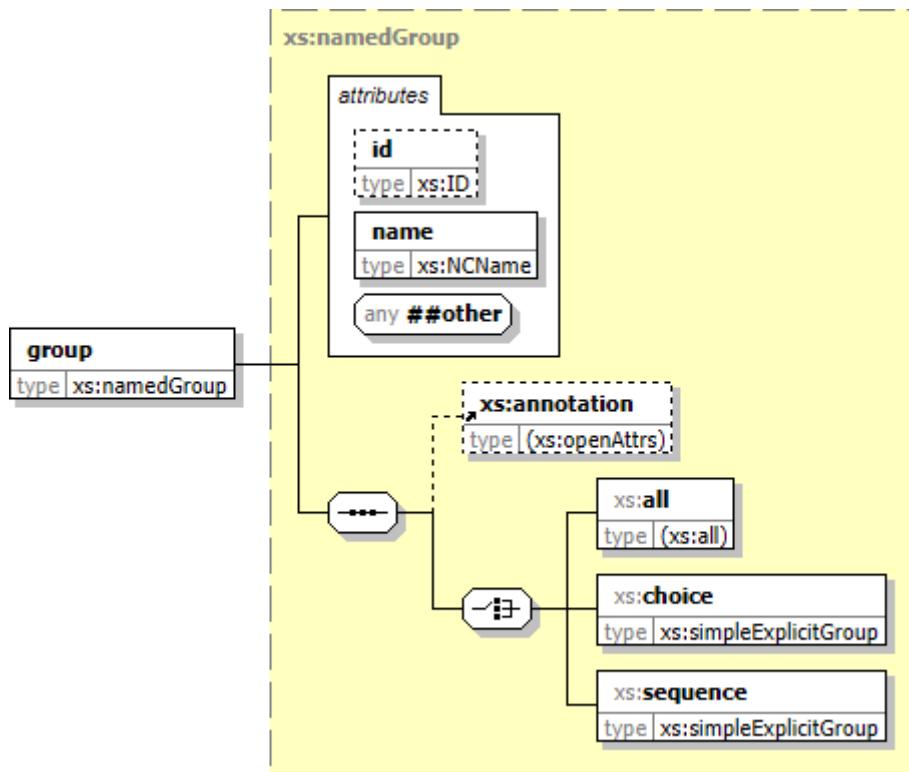
Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

global element

<xs:group>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:namedGroup](#) [212]
 Content: complex, 2 attributes, attr. wildcard, 4 elements
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [78]

Component Diagram



XML Representation Summary

```
<xs:group
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:all | xs:choice | xs:sequence)
</xs:group>
```

Content model elements (4):

[xs:all](#) (in [xs:group](#)) [15], [xs:choice](#) (in [xs:group](#)) [39],
[xs:annotation](#) [17], [xs:sequence](#) (in [xs:group](#)) [131]

Included in content model of elements (2):

[xs:redefine](#) [111], [xs:schema](#) [7]

Known Usage Locations

- Within model groups (1):

[xs:redefinable](#) [354]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-group>

XML Source (see within schema source: p. 408)

```
<xs:element id="group" name="group" type="xs:namedGroup">  
  <xs:annotation>  
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-group"/>  
  </xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [214] within [xs:namedGroup](#) complexType; see [XML source](#) [406]

{any attribute from non-schema namespace}

Defined: [locally](#) [214] within [xs:namedGroup](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 4/4)

xs:all [15]

Type: [anonymous](#) complexType ([restriction of xs:all](#)) [16], complex content
Defined: [locally](#) [214] within [xs:namedGroup](#) complexType; see [XML source](#) [406]

xs:annotation [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [214] within [xs:namedGroup](#) complexType; see [XML source](#) [406]

xs:choice [39]

Type: [xs:simpleExplicitGroup](#) [233], complex content
Defined: [locally](#) [214] within [xs:namedGroup](#) complexType; see [XML source](#) [406]

xs:sequence [131]

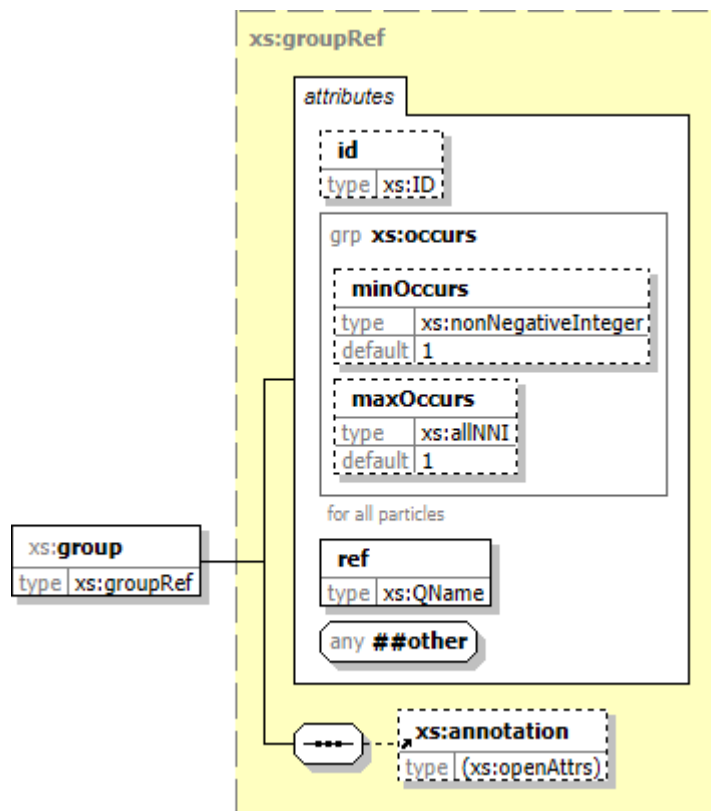
Type: [xs:simpleExplicitGroup](#) [233], complex content
Defined: [locally](#) [214] within [xs:namedGroup](#) complexType; see [XML source](#) [406]

unified local element

`<xs:group>` (type `xs:groupRef`)

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:groupRef](#) [192]
 Content: complex, 4 attributes, attr. wildcard, 1 element
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: locally at 3 locations in [XMLSchema.xsd](#)

Component Diagram



XML Representation Summary

```
<xs:group
  id           = xs:ID
  minOccurs    = xs:nonNegativeInteger : "1"
  maxOccurs    = (xs:nonNegativeInteger | "unbounded") : "1"
  ref          = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:group>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (8):

[xs:choice](#) [36],
[xs:choice](#) (in [xs:group](#)) [39],
[xs:complexType](#) [44],
[xs:complexType](#) (type [xs:localComplexType](#)) [48],
[xs:extension](#) (in [xs:complexContent](#)) [67],
[xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:sequence](#) [128],
[xs:sequence](#) (in [xs:group](#)) [131]

Definition Locations

- Within model groups (3):

[xs:nestedParticle](#) [350], [xs:particle](#) [352], [xs:typeDefParticle](#) [362]

Annotations (1) (by all definition locations)

Locations (3):

within [xs:typeDefParticle](#) group [362], within [xs:nestedParticle](#) group [350], within [xs:particle](#) group [352]

Annotation:

Attribute Detail (all declarations; 5/5)

id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Type: [xs:allNNI](#) [263]

Use: optional

Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

[xs:nonNegativeInteger](#) | "unbounded"

Default: "1"

minOccurs

Type: [xs:nonNegativeInteger](#) [314]

Use: optional

Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

ref

Type: [xs:QName](#) [322]

Use: required

Defined: [locally](#) [194] within [xs:groupRef](#) complexType; see [XML source](#) [406]

{any attribute from non-schema namespace}

Defined: [locally](#) [194] within [xs:groupRef](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content

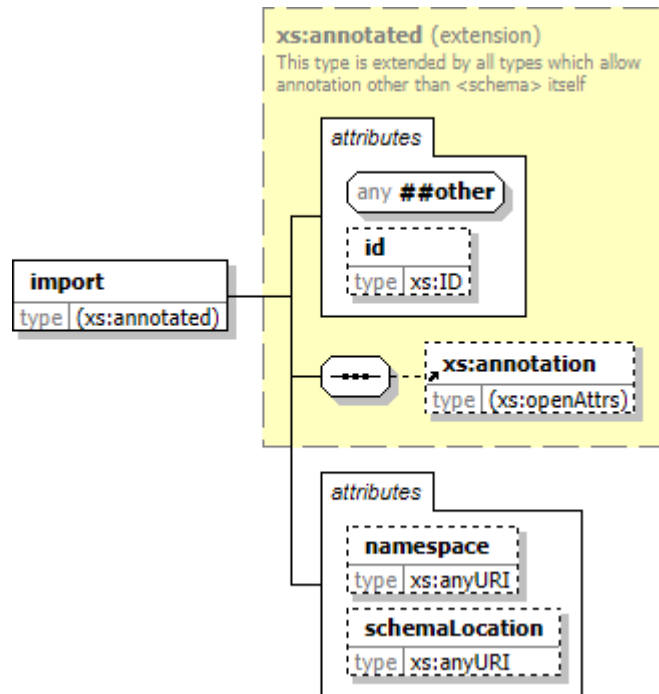
Defined: [by reference](#) [194] within [xs:groupRef](#) complexType; see [XML source](#) [406]

global element

<xs:import>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [82]
Content: complex, 3 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [82]

Component Diagram



XML Representation Summary

```
<xs:import
  id           = xs:ID
  namespace    = xs:anyURI
  schemaLocation = xs:anyURI
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:import>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (1):

[xs:schema](#) [7]

Known Usage Locations

- Within anonymous complexTypes of elements (1):

[xs:schema](#) [11]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-import>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
    
```

XML Source (see within schema source: p. 410)

```

<xs:element id="import" name="import">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-import"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="namespace" type="xs:anyURI"/>
        <xs:attribute name="schemaLocation" type="xs:anyURI"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 4/4)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

namespace

Type: [xs:anyURI](#) [265]
Use: optional
Defined: [locally](#) within [\(this\) xs:import](#) element; see [XML source](#) [410]

schemaLocation

Type: [xs:anyURI](#) [265]
Use: optional
Defined: [locally](#) within [\(this\) xs:import](#) element; see [XML source](#) [410]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

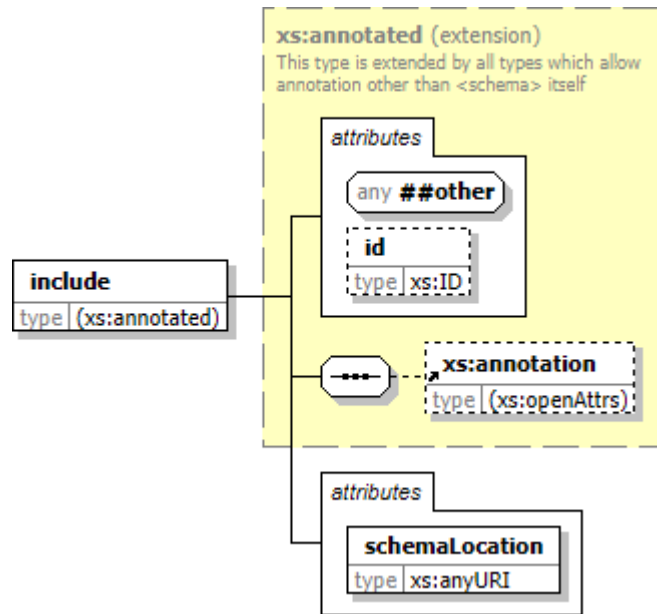
[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element <xs:include>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [84]
Content: complex, 2 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [84]

Component Diagram



XML Representation Summary

```
<xs:include
  id = xs:ID
  schemaLocation = xs:anyURI
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:include>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (1):

[xs:schema](#) [7]

Known Usage Locations

- Within anonymous complexTypes of elements (1):

[xs:schema](#) [11]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-include>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
    
```

XML Source (see within schema source: p. 410)

```

<xs:element id="include" name="include">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-include"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="schemaLocation" type="xs:anyURI" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

schemaLocation

Type: [xs:anyURI](#) [265]
Use: required
Defined: locally within ([this](#)) [xs:include](#) element; see [XML source](#) [410]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

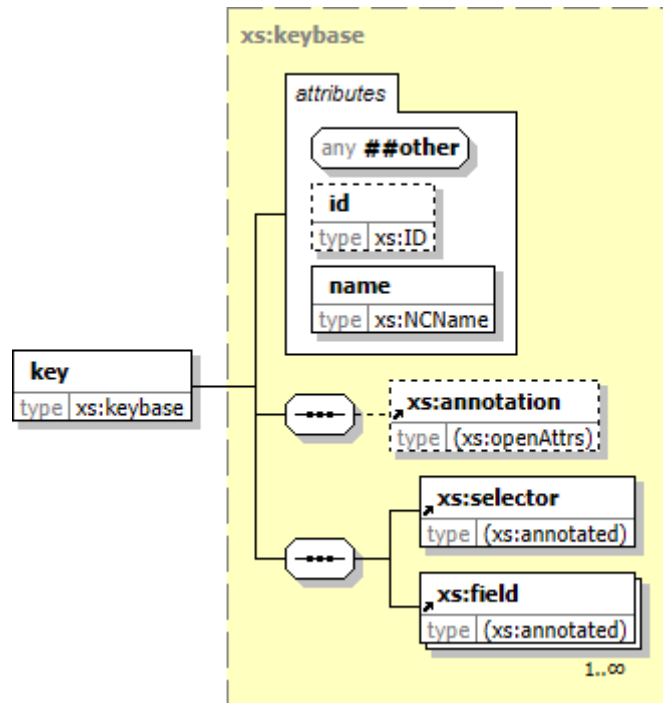
Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:key>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:keybase](#) [195]
 Content: complex, 2 attributes, attr. wildcard, 3 elements
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [86]

Component Diagram



XML Representation Summary

```
<xs:key
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:selector, xs:field+
</xs:key>
```

Content model elements (3):

[xs:annotation](#) [17], [xs:field](#) [72], [xs:selector](#) [125]

Included in content model of elements (3):

[xs:element](#) [53], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]
[xs:element](#) (type [xs:localElement](#)) [57],

Known Usage Locations

- Within model groups (1):
[xs:identityConstraint](#) [347]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-key>

XML Source (see within schema source: p. 412)

```
<xs:element id="key" name="key" type="xs:keybase">  
  <xs:annotation>  
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-key"/>  
  </xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 3/3)

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 3/3)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:field](#) [72]

Type: [anonymous](#) complexType ([extension of](#) [xs:annotated](#)) [73], complex content
Defined: [by reference](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

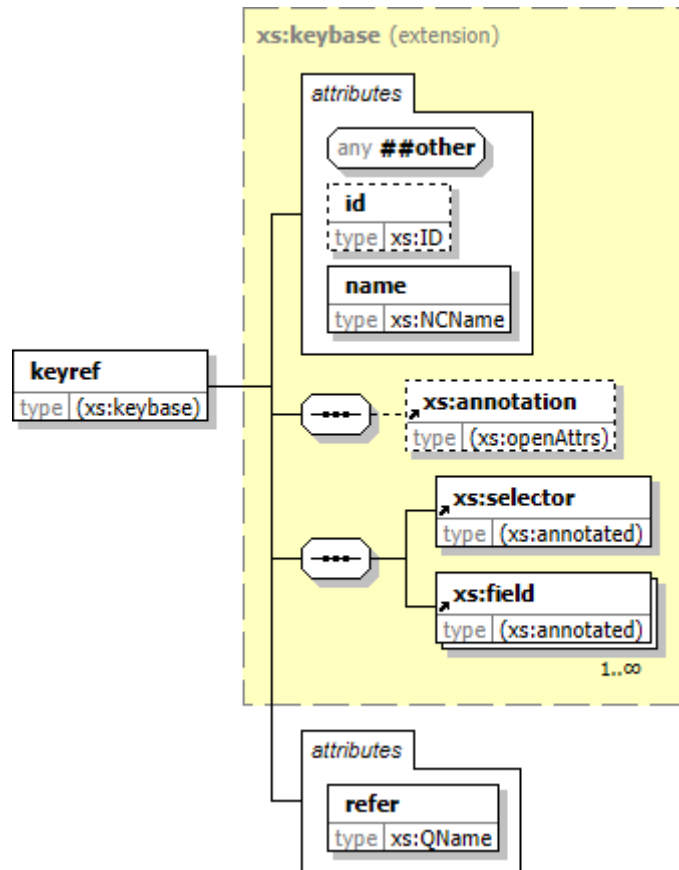
↔ [xs:selector](#) [125]

Type: [anonymous](#) complexType ([extension of](#) [xs:annotated](#)) [126], complex content
Defined: [by reference](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

global element <xs:keyref>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:keybase](#)) [88]
Content: complex, 3 attributes, attr. wildcard, 3 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [88]

Component Diagram



XML Representation Summary

```
<xs:keyref
  id      = xs:ID
  name    = xs:NCName
  refer   = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:selector, xs:field+
</xs:keyref>
```

Content model elements (3):

[xs:annotation](#) [17], [xs:field](#) [72], [xs:selector](#) [125]

Included in content model of elements (3):

[xs:element](#) [53], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]
[xs:element](#) (type [xs:localElement](#)) [57],

Known Usage Locations

- Within model groups (1):
[xs:identityConstraint](#) [348]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-keyref>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:keybase [195] (extension)
│           └── complexType

```

XML Source (see within schema source: p. 412)

```

<xs:element id="keyref" name="keyref">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-keyref"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:keybase">
        <xs:attribute name="refer" type="xs:QName" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 4/4)

- id
 - Type: [xs:ID](#) [295]
 - Use: optional
 - Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]
- name
 - Type: [xs:NCName](#) [308]
 - Use: required
 - Defined: [locally](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]
- refer
 - Type: [xs:QName](#) [322]
 - Use: required
 - Defined: [locally](#) within [\(this\)](#) [xs:keyref](#) element; see [XML source](#) [412]
- {any attribute from non-schema namespace}
 - Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 3/3)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

[xs:field](#) [72]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [73], complex content
Defined: [by reference](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

[xs:selector](#) [125]

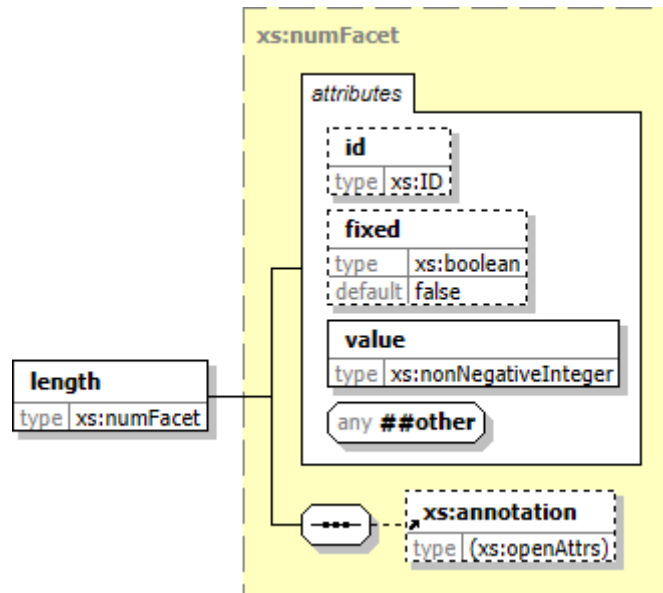
Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [126], complex content
Defined: [by reference](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

global element

`<xs:length>`

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: `xs:numFacet` [222]
Content: complex, 3 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in `XMLSchema.xsd`; see [XML source](#) [91]

Component Diagram



XML Representation Summary

```
<xs:length
  id       = xs:ID
  fixed    = xs:boolean : "false"
  value    = xs:nonNegativeInteger
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:length>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in `xs:simpleContent`) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [345]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-length>

XML Source (see within schema source: p. 426)

```
<xs:element id="length" name="length" type="xs:numFacet">  
  <xs:annotation>  
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-length"/>  
  </xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:nonNegativeInteger](#) [314]
Use: required
Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

{any attribute from non-schema namespace}

Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

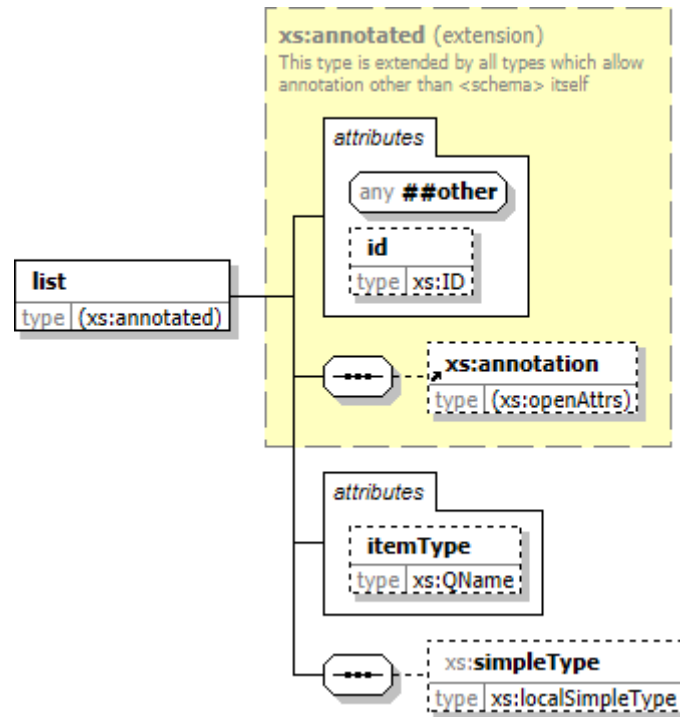
Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

global element

<xs:list>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [93]
Content: complex, 2 attributes, attr. wildcard, 2 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [93]

Component Diagram



XML Representation Summary

```
<xs:list
  id = xs:ID
  itemType = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType?
</xs:list>
```

Content model elements (2):

[xs:annotation](#) [17], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Included in content model of elements (2):

[xs:simpleType](#) [135], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Known Usage Locations

- Within model groups (1):

[xs:simpleDerivation](#) [357]

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
    
```

Annotation

itemType attribute and simpleType child are mutually exclusive, but one or other is required

See: <http://www.w3.org/TR/xmlschema-2/#element-list>

XML Source (see within schema source: p. 425)

```

<xs:element id="list" name="list">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-list">
        itemType attribute and simpleType child are mutually
        exclusive, but one or other is required
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:sequence>
          <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
        </xs:sequence>
        <xs:attribute name="itemType" type="xs:QName" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

itemType

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) within (this) [xs:list](#) element; see [XML source](#) [425]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 2/2)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

[xs:simpleType](#) [138]

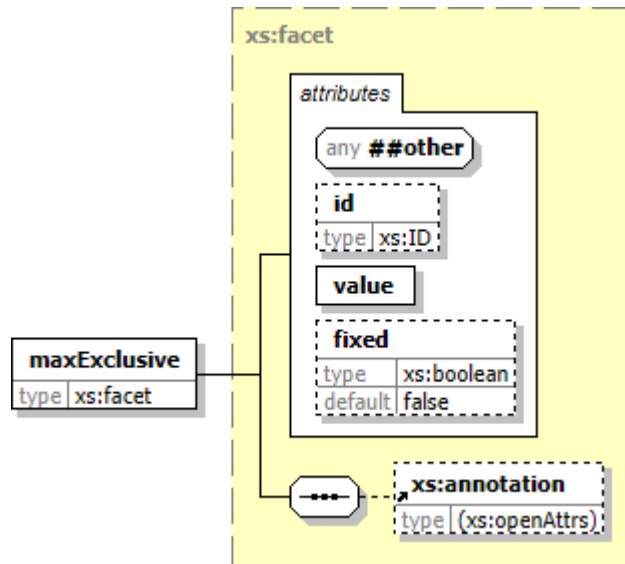
Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) within (this) [xs:list](#) element; see [XML source](#) [425]

global element

<xs:maxExclusive>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:facet](#) [186]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 1 [element](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [94]

Component Diagram



XML Representation Summary

```

<xs:maxExclusive
  id      = xs:ID
  value   = xs:anySimpleType
  fixed   = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:maxExclusive>
    
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [345]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-maxExclusive>

XML Source (see within schema source: p. 426)

```
<xs:element id="maxExclusive" name="maxExclusive" type="xs:facet">
```

```
<xs:annotation>  
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-maxExclusive"/>  
</xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:anySimpleType](#)
Use: required
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

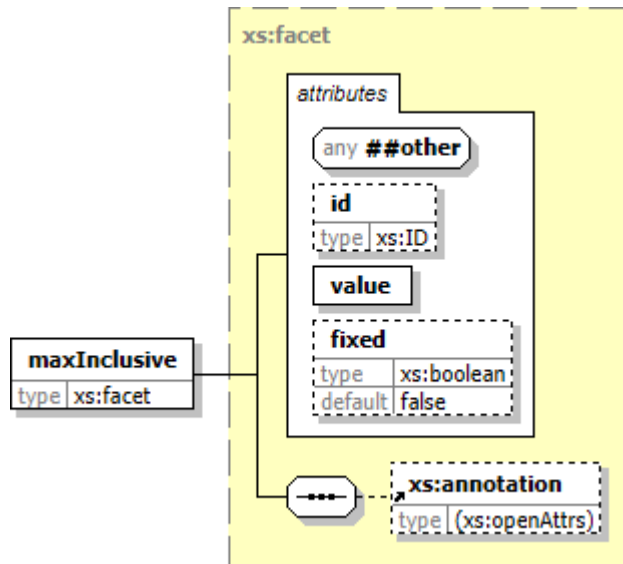
Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:maxInclusive>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:facet](#) [186]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 1 [element](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [96]

Component Diagram



XML Representation Summary

```

<xs:maxInclusive
  id      = xs:ID
  value   = xs:anySimpleType
  fixed   = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?
</xs:maxInclusive>
    
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [345]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-maxInclusive>

XML Source (see within schema source: p. 426)

```
<xs:element id="maxInclusive" name="maxInclusive" type="xs:facet">
```

```
<xs:annotation>  
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-maxInclusive"/>  
</xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:anySimpleType](#)
Use: required
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

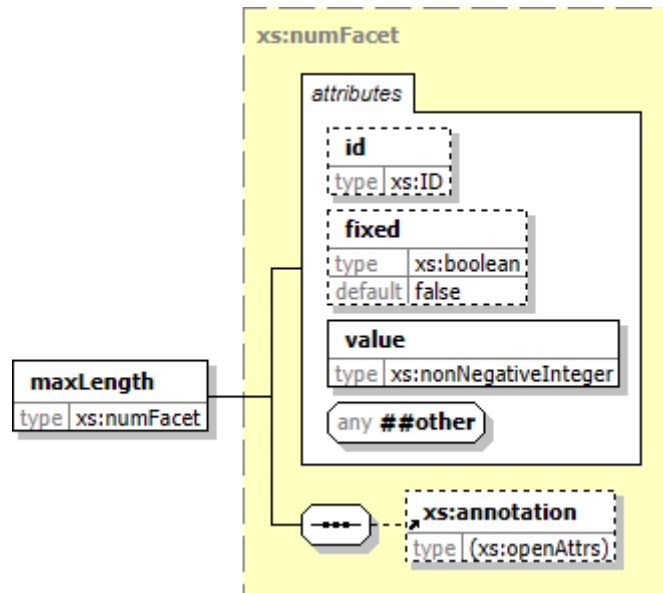
Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:maxLength>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:numFacet](#) [222]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 1 [element](#)
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [99]

Component Diagram



XML Representation Summary

```
<xs:maxLength
  id = xs:ID
  fixed = xs:boolean : "false"
  value = xs:nonNegativeInteger
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:maxLength>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-maxLength>

XML Source (see within schema source: p. 426)

```
<xs:element id="maxLength" name="maxLength" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-maxLength"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:nonNegativeInteger](#) [314]
Use: required
Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

{any attribute from non-schema namespace}

Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

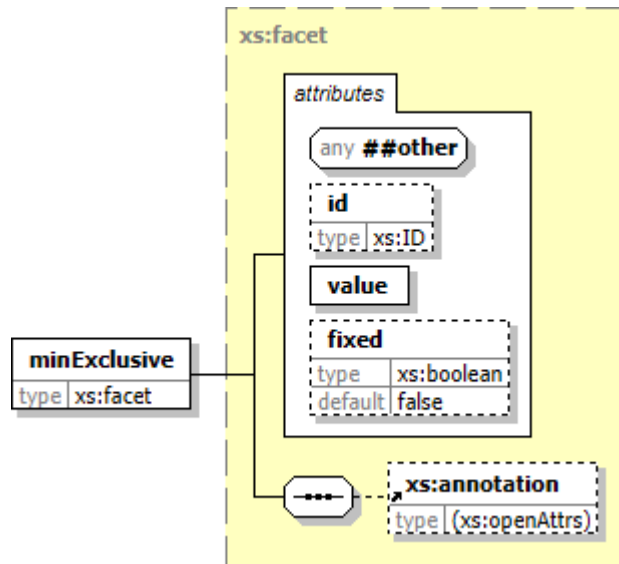
Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

global element

<xs:minExclusive>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:facet](#) [186]
 Content: complex, 3 attributes, attr. wildcard, 1 element
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [100]

Component Diagram



XML Representation Summary

```
<xs:minExclusive
  id      = xs:ID
  value   = xs:anySimpleType
  fixed   = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:minExclusive>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-minExclusive>

XML Source (see within schema source: p. 425)

```
<xs:element id="minExclusive" name="minExclusive" type="xs:facet">
```



```
<xs:annotation>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-minExclusive"/>
</xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

■ fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ value

Type: [xs:anySimpleType](#)
Use: required
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

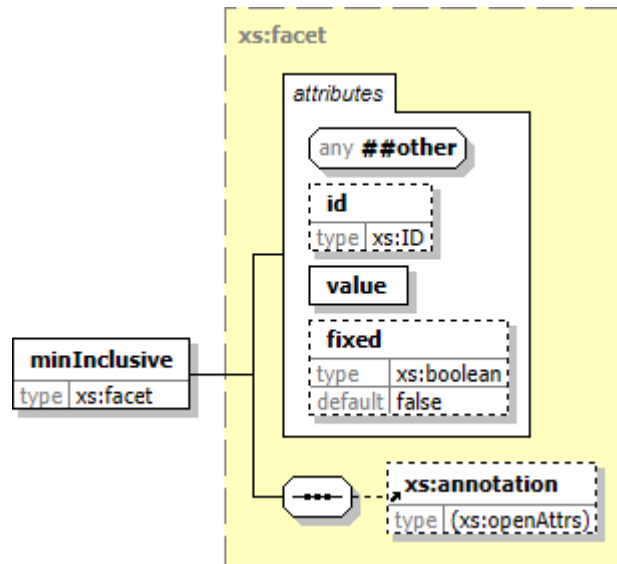
Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:minInclusive>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:facet](#) [186]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 1 [element](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [102]

Component Diagram



XML Representation Summary

```
<xs:minInclusive
  id      = xs:ID
  value  = xs:anySimpleType
  fixed  = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:minInclusive>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-minInclusive>

XML Source (see within schema source: p. 425)

```
<xs:element id="minInclusive" name="minInclusive" type="xs:facet">
```

```
<xs:annotation>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-minInclusive"/>
</xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:anySimpleType](#)
Use: required
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

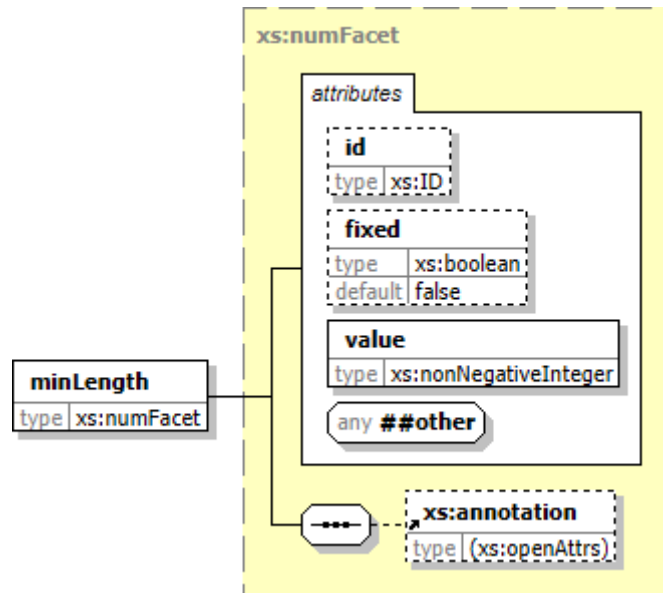
Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:minLength>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:numFacet](#) [222]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 1 [element](#)
Block: "#all" (*blocks all substitutions of this element or its type*)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [105]

Component Diagram



XML Representation Summary

```
<xs:minLength
  id      = xs:ID
  fixed  = xs:boolean : "false"
  value  = xs:nonNegativeInteger
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:minLength>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-minLength>

XML Source (see within schema source: p. 426)

```
<xs:element id="minLength" name="minLength" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-minLength"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:nonNegativeInteger](#) [314]
Use: required
Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

{any attribute from non-schema namespace}

Defined: [locally](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

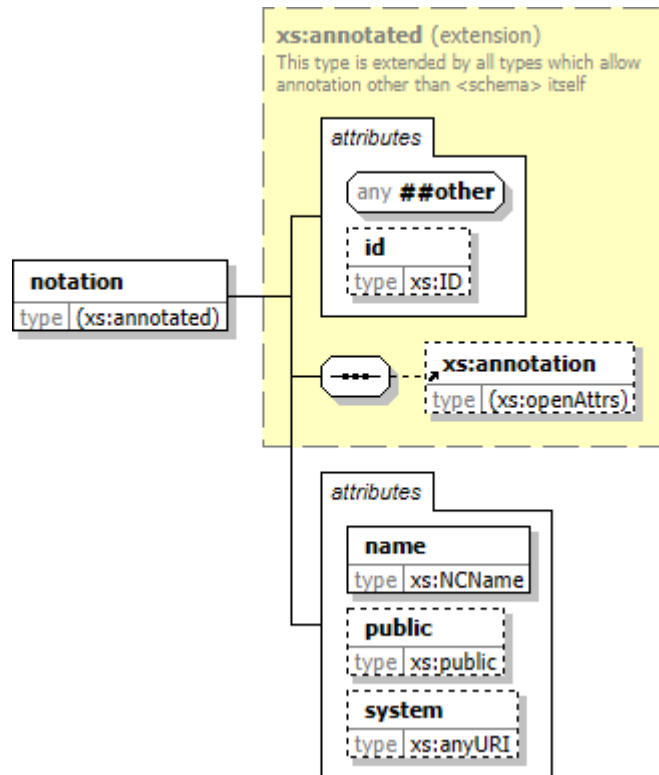
Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [223] within [xs:numFacet](#) complexType; see [XML source](#) [426]

global element

<xs:notation>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [107]
Content: complex, 4 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [107]

Component Diagram



XML Representation Summary

```
<xs:notation
  id      = xs:ID
  name    = xs:NCName
  public  = xs:token
  system  = xs:anyURI
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:notation>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (1):

[xs:schema](#) [7]

Known Usage Locations

- Within model groups (1):

[xs:schemaTop](#) [356]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-notation>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
    
```

XML Source (see within schema source: p. 412)

```

<xs:element id="notation" name="notation">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-notation"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="name" type="xs:NCName" use="required"/>
        <xs:attribute name="public" type="xs:public"/>
        <xs:attribute name="system" type="xs:anyURI"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 5/5)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: locally within (this) [xs:notation](#) element; see [XML source](#) [412]

public

Type: [xs:public](#) [321]
Use: optional
Defined: locally within (this) [xs:notation](#) element; see [XML source](#) [412]

Attribute Value

```
xs:token
```


system

Type: [xs:anyURI](#) [265]
Use: optional
Defined: locally within (this) [xs:notation](#) element; see [XML source](#) [412]

{any attribute from non-schema namespace}

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

 [xs:annotation](#) [17]

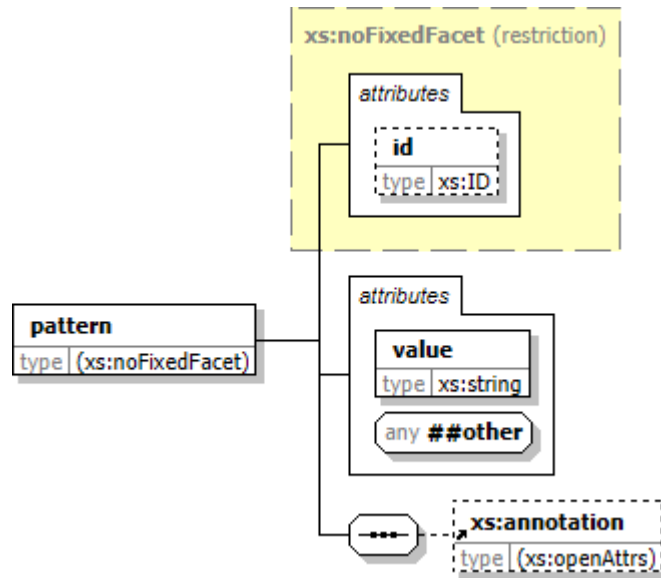
Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

global element

<xs:pattern>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (restriction of [xs:noFixedFacet](#)) [110]
Content: complex, 2 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [110]

Component Diagram



XML Representation Summary

```
<xs:pattern
  id = xs:ID
  value = xs:string
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:pattern>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-pattern>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   ├── xs:annotated [153] (extension)
│   │   ├── xs:facet [186] (restriction)
│   │   │   └── xs:noFixedFacet [220] (restriction)
│   │   └── complexType
│   └──
└──

```

XML Source (see within schema source: p. 427)

```

<xs:element id="pattern" name="pattern">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-pattern"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:noFixedFacet">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="value" type="xs:string" use="required"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:string](#) [328]
Use: required
Defined: locally within (this) [xs:pattern](#) element; see [XML source](#) [427]

{any attribute from non-schema namespace}

Defined: locally within (this) [xs:pattern](#) element; see [XML source](#) [427]

Content Element Detail (all declarations; 1/1)

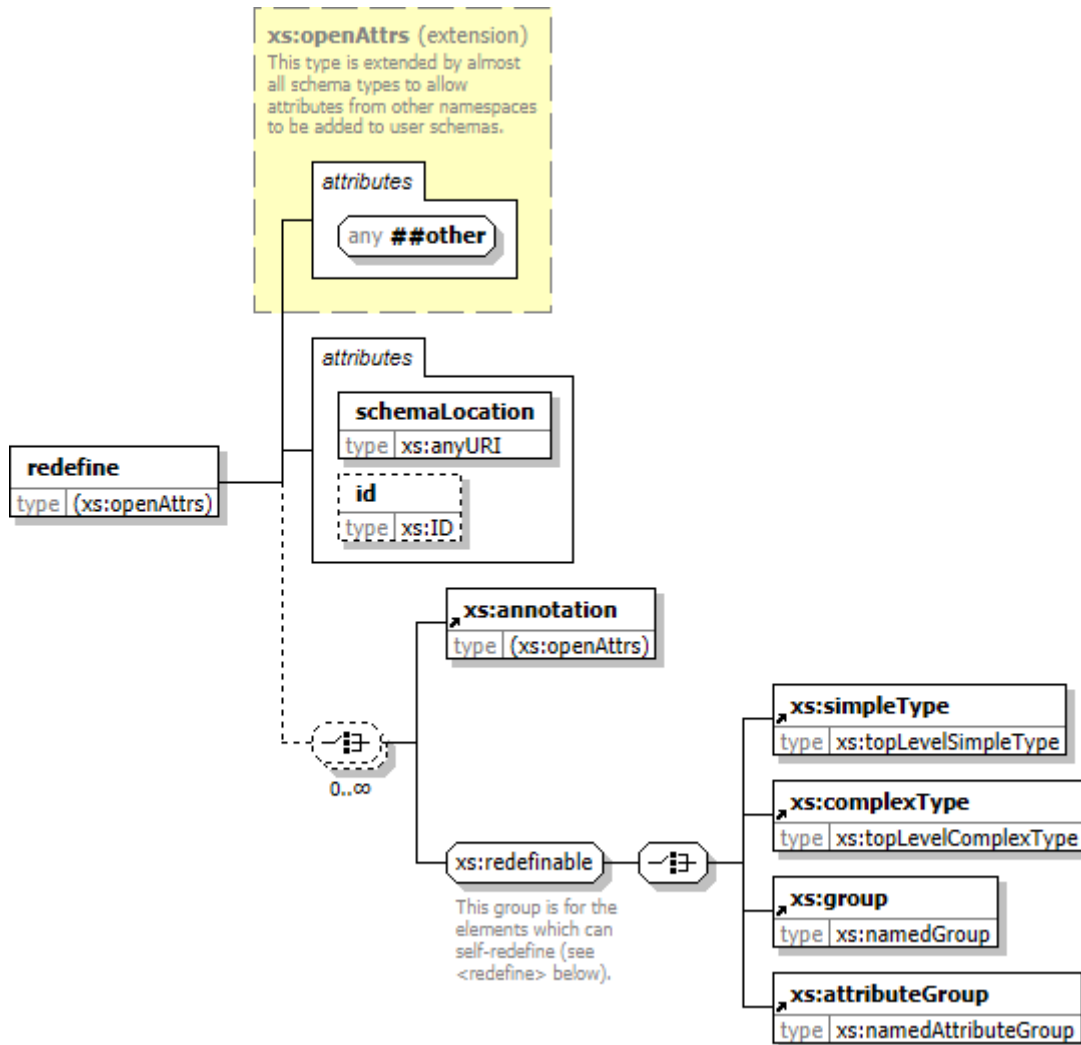
[xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference within (this) [xs:pattern](#) element; see [XML source](#) [427]

global element <xs:redefine>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:openAttrs](#)) [112]
Content: complex, 2 attributes, attr. wildcard, 5 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [112]

Component Diagram



XML Representation Summary

```
<xs:redefine
  schemaLocation = xs:anyURI
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: (xs:annotation | xs:simpleType | xs:complexType | xs:group | xs:attributeGroup)*
</xs:redefine>
```

Content model elements (5):

[xs:annotation](#) [17], [xs:attributeGroup](#) [32], [xs:complexType](#) [44], [xs:group](#) [77], [xs:simpleType](#) [135]

Included in content model of elements (1):

[xs:schema](#) [7]

Known Usage Locations

- Within anonymous complexTypes of elements (1):
[xs:schema](#) [11]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-redefine>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── complexType

```

XML Source (see within schema source: p. 410)

```

<xs:element id="redefine" name="redefine">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-redefine"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:choice maxOccurs="unbounded" minOccurs="0">
          <xs:element ref="xs:annotation"/>
          <xs:group ref="xs:redefinable"/>
        </xs:choice>
        <xs:attribute name="schemaLocation" type="xs:anyURI" use="required"/>
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally within (this) [xs:redefine](#) element; see [XML source](#) [410]

schemaLocation

Type: [xs:anyURI](#) [265]
Use: required
Defined: locally within (this) [xs:redefine](#) element; see [XML source](#) [410]

{any attribute from non-schema namespace}

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 5/5)

↔ [xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference within (this) [xs:redefine](#) element; see [XML source](#) [410]

↔ `xs:attributeGroup` [32]

Type: `xs:namedAttributeGroup` [209], complex content
Defined: [by reference](#) [354] within `xs:redefinable` group; see [XML source](#) [398]

↔ `xs:complexType` [44]

Type: `xs:topLevelComplexType` [249], complex content
Defined: [by reference](#) [354] within `xs:redefinable` group; see [XML source](#) [398]

↔ `xs:group` [77]

Type: `xs:namedGroup` [212], complex content
Defined: [by reference](#) [354] within `xs:redefinable` group; see [XML source](#) [398]

↔ `xs:simpleType` [135]

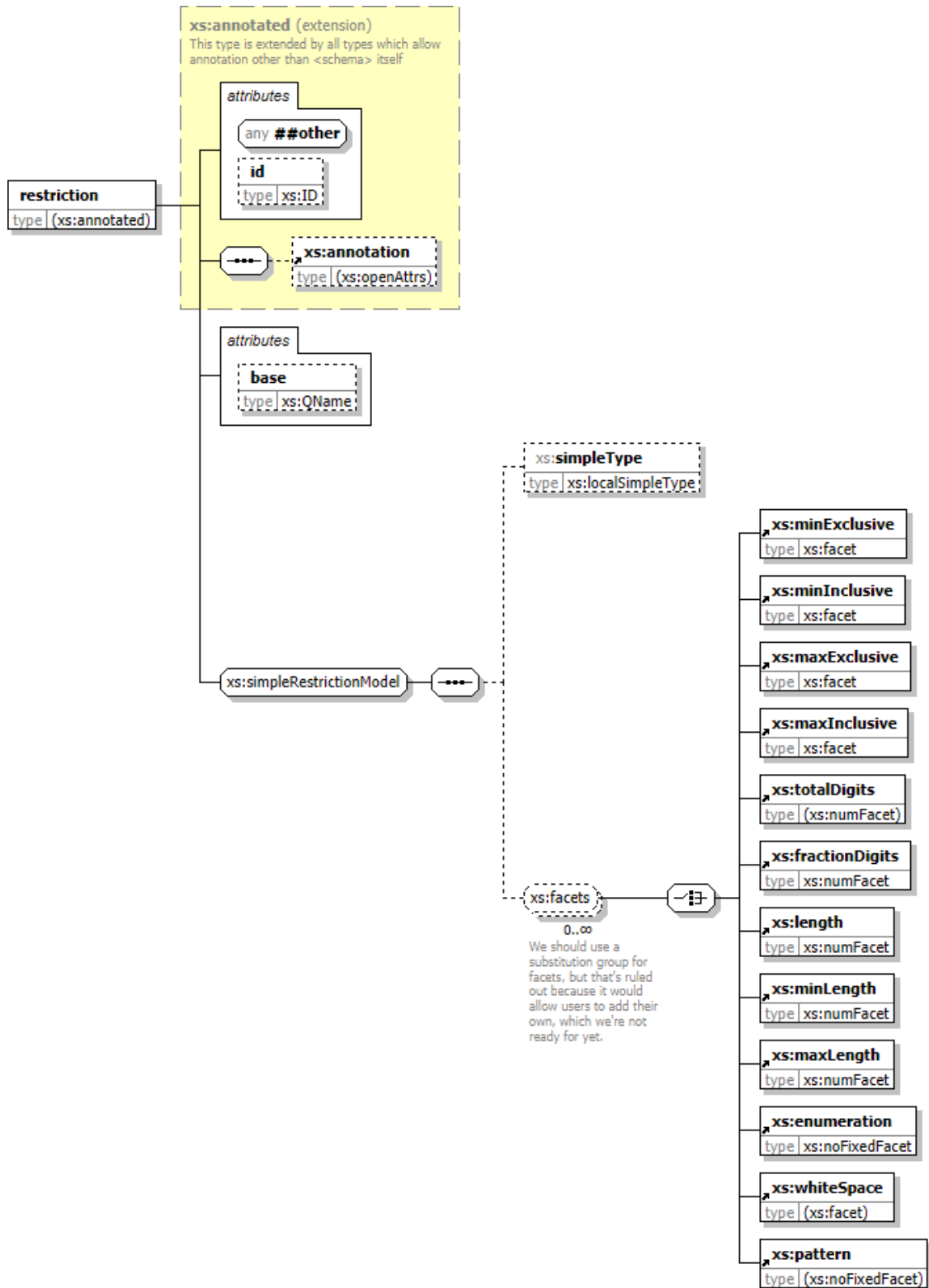
Type: `xs:topLevelSimpleType` [257], complex content
Defined: [by reference](#) [354] within `xs:redefinable` group; see [XML source](#) [398]

global element

<xs:restriction>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of `xs:annotated`) [115]
Content: complex, 2 attributes, attr. wildcard, 14 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in `XMLSchema.xsd`; see [XML source](#) [115]

Component Diagram



XML Representation Summary

```
<xs:restriction
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType?, (xs:minExclusive | xs:minInclusive | xs:maxExclusive |
      xs:maxInclusive | xs:totalDigits | xs:fractionDigits | xs:length | xs:minLength | xs:maxLength |
      xs:enumeration | xs:whiteSpace | xs:pattern)*
</xs:restriction>
```

Content model elements (14):

xs:annotation [17],	xs:minExclusive [100],
xs:enumeration [65],	xs:minInclusive [102],
xs:fractionDigits [75],	xs:minLength [104],
xs:length [90],	xs:pattern [109],
xs:maxExclusive [94],	xs:simpleType (type xs:localSimpleType) [138],
xs:maxInclusive [96],	xs:totalDigits [140],
xs:maxLength [98],	xs:whiteSpace [147]

Included in content model of elements (2):

xs:simpleType [135], xs:simpleType (type xs:localSimpleType) [138]

Known Usage Locations

- Within model groups (1):

xs:simpleDerivation [357]

Anonymous Type Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
```

Annotation

base attribute and simpleType child are mutually exclusive, but one or other is required

See: <http://www.w3.org/TR/xmlschema-2/#element-restriction>

XML Source (see within schema source: p. 424)

```
<xs:element id="restriction" name="restriction">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-restriction">
        base attribute and simpleType child are mutually
        exclusive, but one or other is required
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:group ref="xs:simpleRestrictionModel"/>
        <xs:attribute name="base" type="xs:QName" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Attribute Detail (all declarations; 3/3)

■ base

Type: [xs:QName](#) [322]
Use: optional
Defined: locally within ([this](#)) [xs:restriction](#) element; see [XML source](#) [425]

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 14/14)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:enumeration](#) [65]

Type: [xs:noFixedFacet](#) [220], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:fractionDigits](#) [75]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:length](#) [90]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxExclusive](#) [94]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxInclusive](#) [96]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxLength](#) [98]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minExclusive](#) [100]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minInclusive](#) [102]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minLength](#) [104]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:pattern](#) [109]

Type: [anonymous complexType \(restriction of xs:noFixedFacet\)](#) [110], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) [360] within [xs:simpleRestrictionModel](#) group; see [XML source](#) [424]

↔ [xs:totalDigits](#) [140]

Type: [anonymous complexType \(restriction of xs:numFacet\)](#) [141], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:whiteSpace](#) [147]

Type: [anonymous complexType \(restriction of xs:facet\)](#) [148], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

local element

`<xs:restriction>` (in `xs:complexContent`)

Namespace: <http://www.w3.org/2001/XMLSchema>

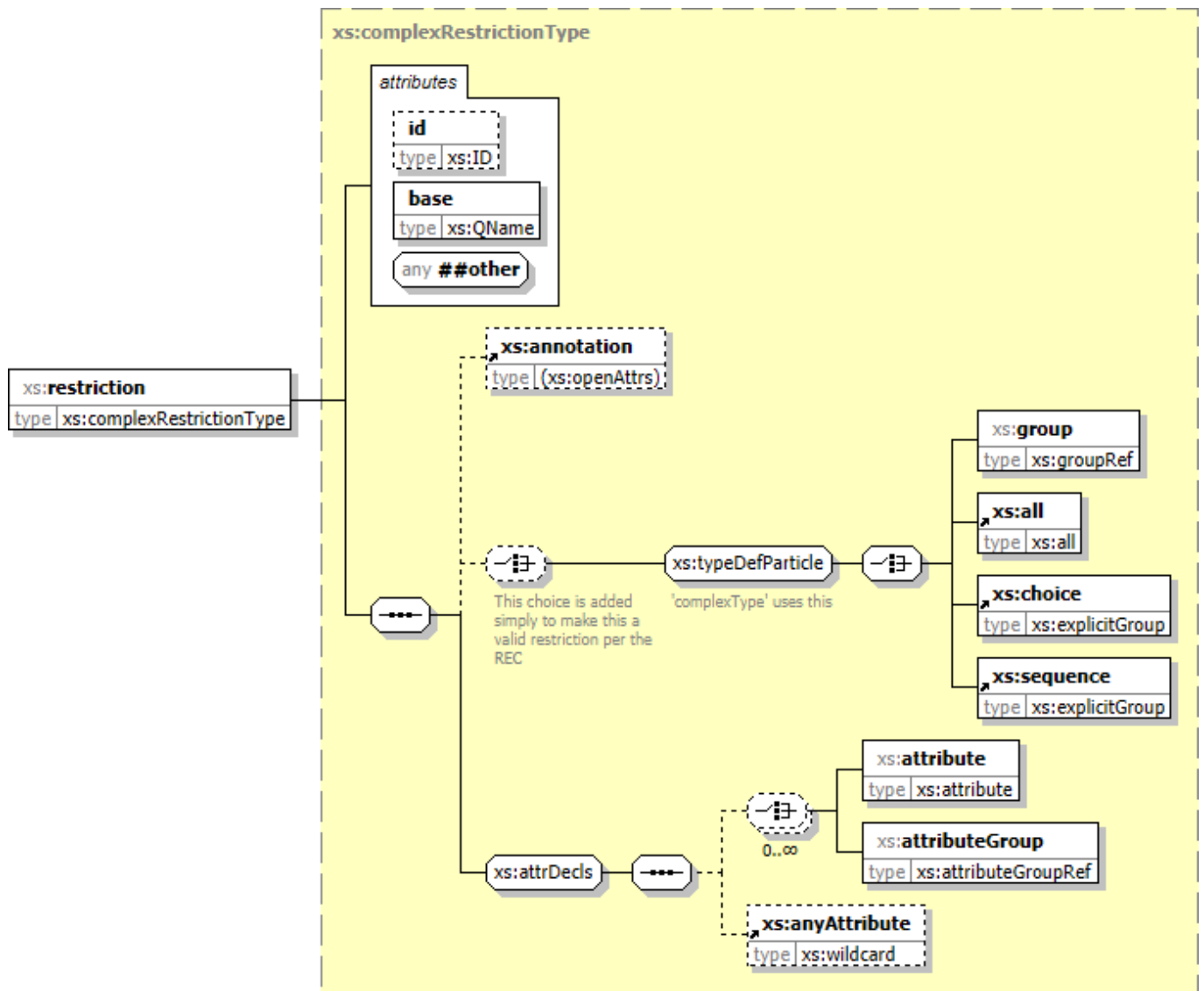
Type: `xs:complexRestrictionType` [167]

Content: complex, 2 attributes, attr. wildcard, 8 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within `xs:complexContent` element [43] in `XMLSchema.xsd`; see [XML source](#) [119]

Component Diagram



XML Representation Summary

```
<xs:restriction
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:group | xs:all | xs:choice | xs:sequence)?, (xs:attribute |
  xs:attributeGroup)*, xs:anyAttribute?
</xs:restriction>
```

Content model elements (8):

`xs:all` [12],

`xs:attributeGroup` (type `xs:attributeGroupRef`) [34],

[xs:annotation](#) [17], [xs:choice](#) [36],
[xs:anyAttribute](#) [23], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:attribute](#) (type [xs:attribute](#)) [29], [xs:sequence](#) [128]

Included in content model of elements (1):

[xs:complexContent](#) [41]

XML Source (see within schema source: p. 403)

```
<xs:element name="restriction" type="xs:complexRestrictionType"/>
```

Attribute Detail (all declarations; 3/3)

base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [231] within [xs:restrictionType](#) complexType; see [XML source](#) [402]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: [locally](#) [169] within [xs:complexRestrictionType](#) complexType; see [XML source](#) [403]

Content Element Detail (all declarations; 8/8)

↔ [xs:all](#) [12]

Type: [xs:all](#) [150], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [169] within [xs:complexRestrictionType](#) complexType; see [XML source](#) [402]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

 [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

local element

`<xs:restriction>` (in `xs:simpleContent`)

Namespace: <http://www.w3.org/2001/XMLSchema>

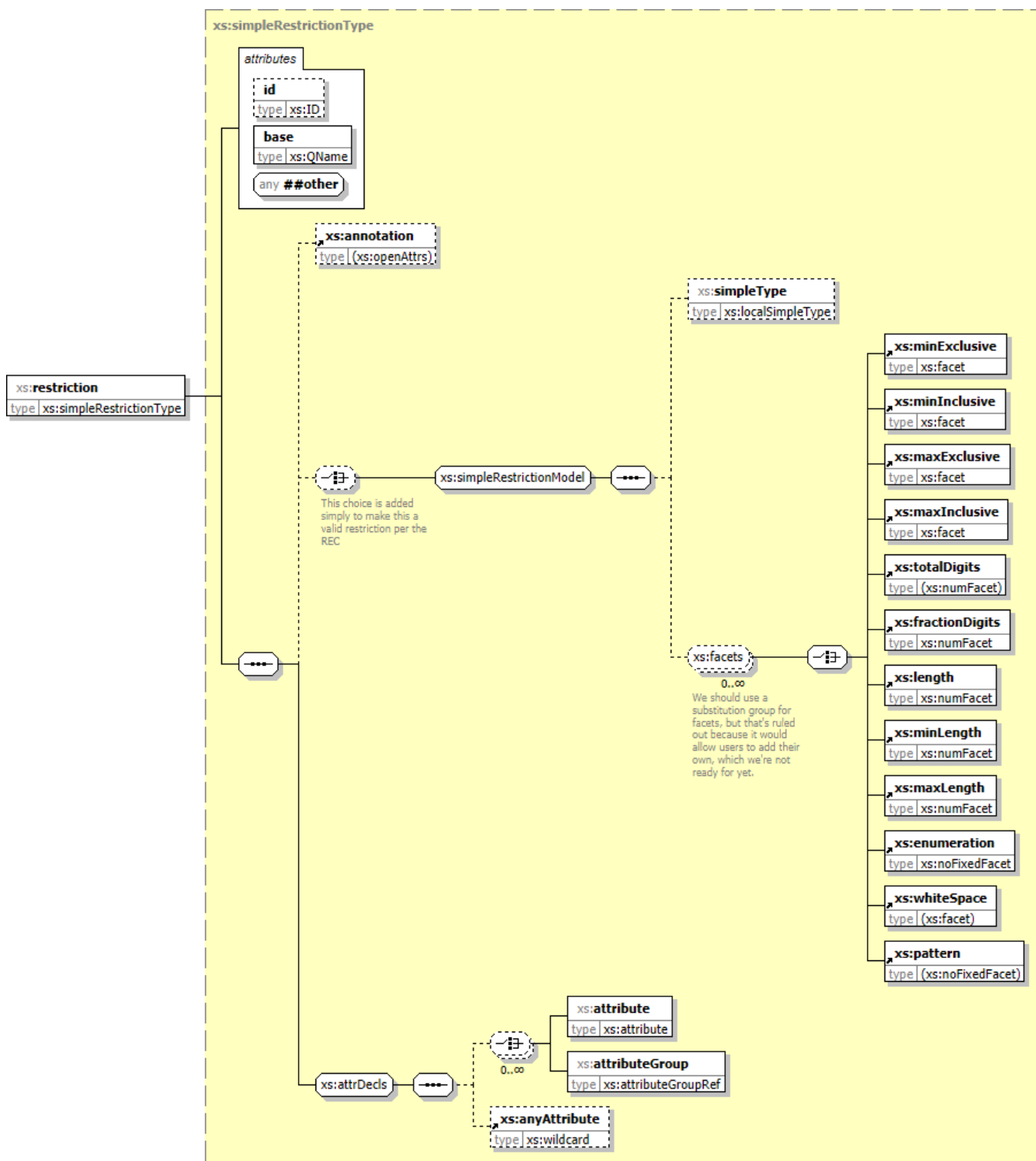
Type: `xs:simpleRestrictionType` [239]

Content: complex, 2 attributes, attr. `wildcard`, 17 elements

Block: "#all" (blocks all substitutions of this element or its type)

Defined: locally within `xs:simpleContent` element [134] in `XMLSchema.xsd`; see [XML source](#) [122]

Component Diagram



XML Representation Summary

```

<xs:restriction
  id       = xs:ID
  base     = xs:QName
  {any attribute from non-schema namespace}
>
Content:  xs:annotation?, (xs:simpleType?, (xs:minExclusive | xs:minInclusive | xs:maxExclusive |
      xs:maxInclusive | xs:totalDigits | xs:fractionDigits | xs:length | xs:minLength | xs:maxLength |
      xs:enumeration | xs:whiteSpace | xs:pattern)*)?, (xs:attribute | xs:attributeGroup)*,
      xs:anyAttribute?
</xs:restriction>
    
```

Content model elements (17):

xs:annotation [17],	xs:maxLength [98],
xs:anyAttribute [23],	xs:minExclusive [100],
xs:attribute (type xs:attribute) [29],	xs:minInclusive [102],
xs:attributeGroup (type xs:attributeGroupRef) [34],	xs:minLength [104],
xs:enumeration [65],	xs:pattern [109],
xs:fractionDigits [75],	xs:simpleType (type xs:localSimpleType) [138],
xs:length [90],	xs:totalDigits [140],
xs:maxExclusive [94],	xs:whiteSpace [147]
xs:maxInclusive [96],	

Included in content model of elements (1):

[xs:simpleContent](#) [133]

XML Source (see within schema source: p. 404)

```
<xs:element name="restriction" type="xs:simpleRestrictionType"/>
```

Attribute Detail (all declarations; 3/3)

base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [231] within [xs:restrictionType](#) complexType; see [XML source](#) [402]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: [locally](#) [241] within [xs:simpleRestrictionType](#) complexType; see [XML source](#) [403]

Content Element Detail (all declarations; 17/17)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [241] within [xs:simpleRestrictionType](#) complexType; see [XML source](#) [403]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ `xs:attribute` [29]

Type: `xs:attribute` [158], complex content
Defined: locally [340] within `xs:attrDecls` group; see [XML source](#) [401]

↔ `xs:attributeGroup` [34]

Type: `xs:attributeGroupRef` [165], complex content
Defined: locally [341] within `xs:attrDecls` group; see [XML source](#) [401]

↔ `xs:enumeration` [65]

Type: `xs:noFixedFacet` [220], complex content
Defined: by reference [345] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:fractionDigits` [75]

Type: `xs:numFacet` [222], complex content
Defined: by reference [345] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:length` [90]

Type: `xs:numFacet` [222], complex content
Defined: by reference [345] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:maxExclusive` [94]

Type: `xs:facet` [186], complex content
Defined: by reference [345] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:maxInclusive` [96]

Type: `xs:facet` [186], complex content
Defined: by reference [345] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:maxLength` [98]

Type: `xs:numFacet` [222], complex content
Defined: by reference [346] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:minExclusive` [100]

Type: `xs:facet` [186], complex content
Defined: by reference [346] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:minInclusive` [102]

Type: `xs:facet` [186], complex content
Defined: by reference [346] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:minLength` [104]

Type: `xs:numFacet` [222], complex content
Defined: by reference [346] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:pattern` [109]

Type: anonymous complexType (restriction of `xs:noFixedFacet`) [110], complex content
Defined: by reference [346] within `xs:facets` group; see [XML source](#) [424]

↔ `xs:simpleType` [138]

Type: `xs:localSimpleType` [206], complex content
Defined: locally [360] within `xs:simpleRestrictionModel` group; see [XML source](#) [424]

↔ [xs:totalDigits](#) [140]

Type: anonymous complexType (restriction of [xs:numFacet](#)) [141], complex content
Defined: by reference [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:whiteSpace](#) [147]

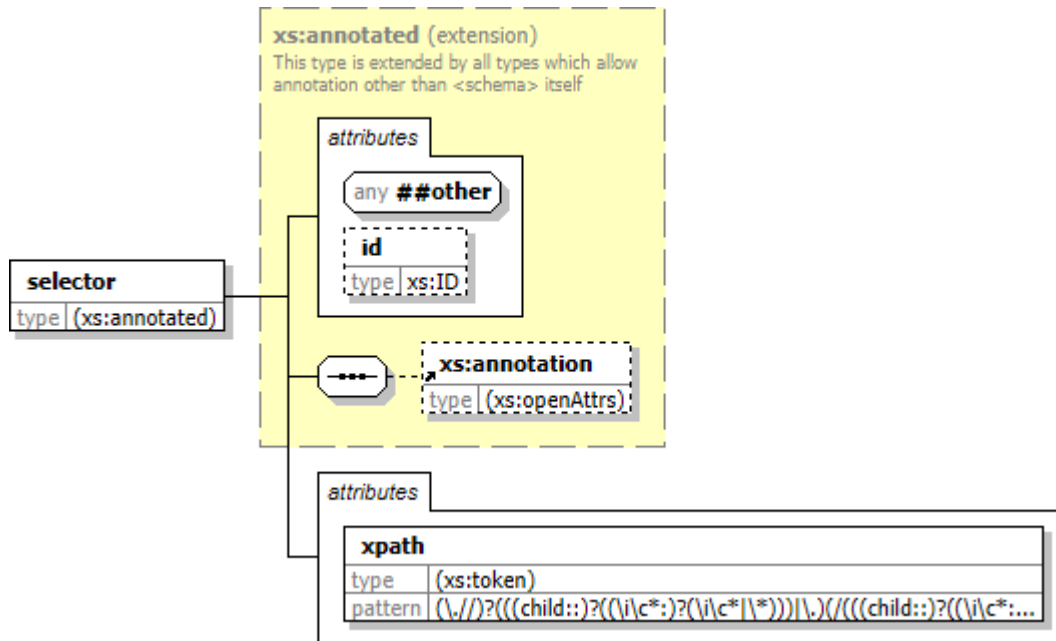
Type: anonymous complexType (restriction of [xs:facet](#)) [148], complex content
Defined: by reference [346] within [xs:facets](#) group; see [XML source](#) [424]

global element

<xs:selector>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: **anonymous** complexType (extension of [xs:annotated](#)) [126]
 Content: complex, 2 **attributes**, attr. **wildcard**, 1 **element**
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [126]

Component Diagram



XML Representation Summary

```
<xs:selector
  id = xs:ID
  xpath = xs:token
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:selector>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (3):

[xs:key](#) [85], [xs:keyref](#) [87], [xs:unique](#) [145]

Known Usage Locations

- Within global complexTypes (1):

[xs:keybase](#) [196]

Annotation

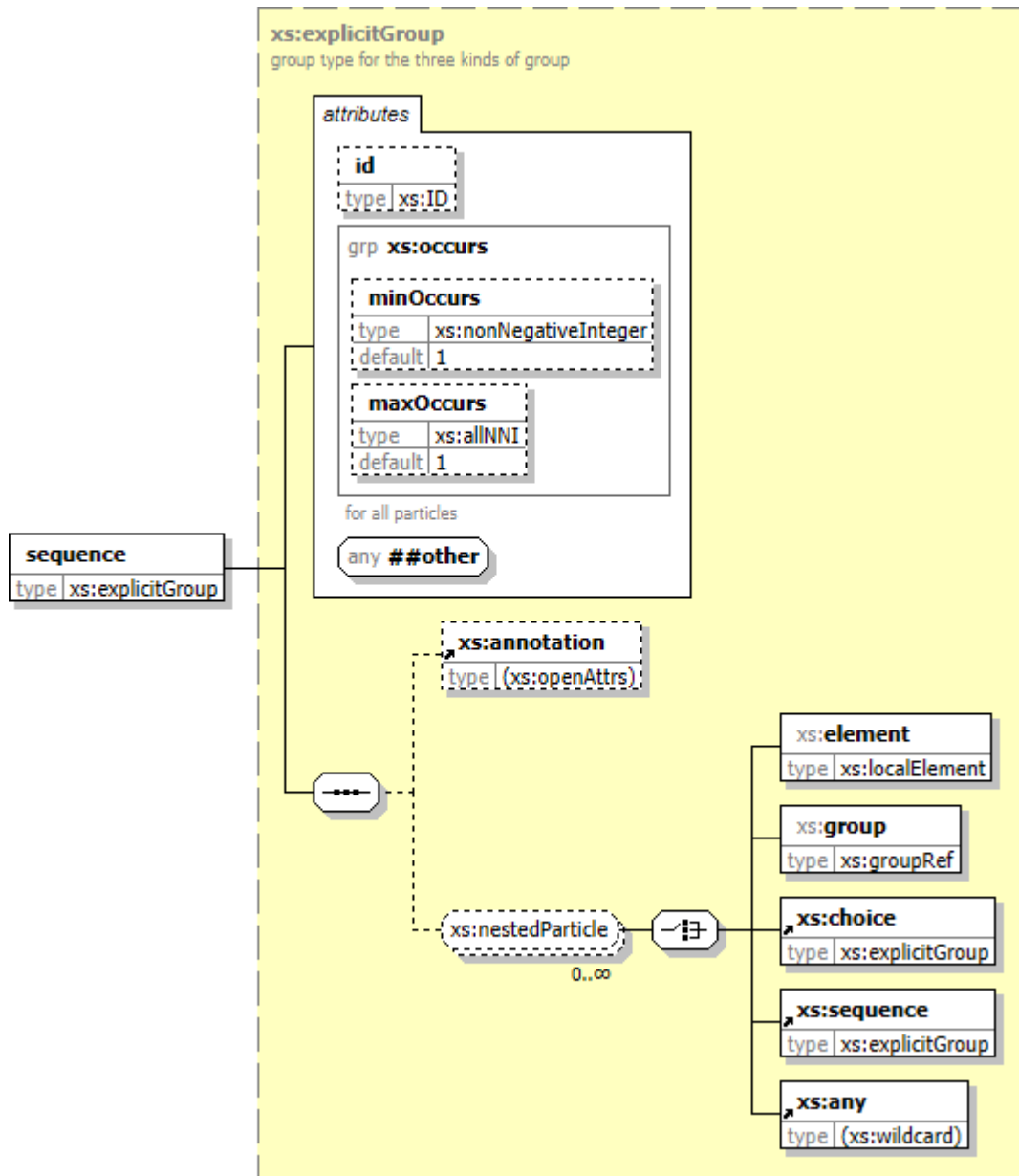
See: <http://www.w3.org/TR/xmlschema-1/#element-selector>

global element

<xs:sequence>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:explicitGroup](#) [179]
Content: complex, 3 [attributes](#), attr. [wildcard](#), 6 [elements](#)
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [129]

Component Diagram



XML Representation Summary

```
<xs:sequence
  id = xs:ID
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:element | xs:group | xs:choice | xs:sequence | xs:any)*
</xs:sequence>
```

Content model elements (6):

[xs:annotation](#) [17], [xs:element](#) (type [xs:localElement](#)) [57],
[xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]

Included in content model of elements (8):

[xs:choice](#) [36], [xs:extension](#) (in [xs:complexContent](#)) [67],
[xs:choice](#) (in [xs:group](#)) [39], [xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:complexType](#) [44], [xs:sequence](#) [128],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:sequence](#) (in [xs:group](#)) [131]

Known Usage Locations

- **Within global complexTypes (1):**
[xs:realGroup](#) [228]
- **Within model groups (3):**
[xs:nestedParticle](#) [350], [xs:particle](#) [352], [xs:typeDefParticle](#) [362]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>

XML Source (see within schema source: p. 408)

```
<xs:element id="sequence" name="sequence" type="xs:explicitGroup">  
  <xs:annotation>  
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-sequence"/>  
  </xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 4/4)

■ id
Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs
Type: [xs:allNNI](#) [263]
Use: optional
Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

■ minOccurs
Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

■ {any attribute from non-schema namespace}

Defined: [locally](#) [181] within [xs:explicitGroup](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) [181] within [xs:explicitGroup](#) complexType; see [XML source](#) [406]

↔ [xs:any](#) [20]

Type: [anonymous](#) complexType (extension of [xs:wildcard](#)) [21], complex content

Defined: [by reference](#) [349] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

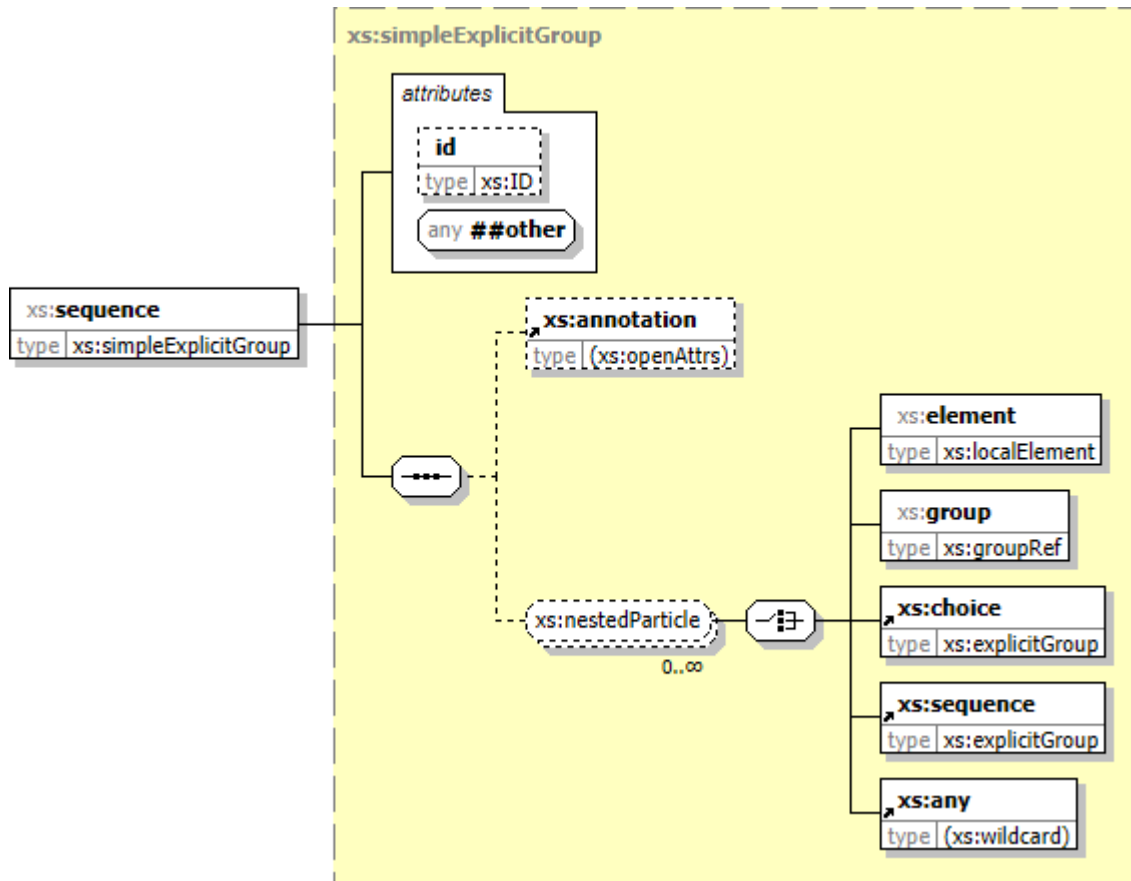
Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

local element

<xs:sequence> (in xs:group)

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:simpleExplicitGroup](#) [233]
Content: complex, 1 attribute, attr. wildcard, 6 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: locally within [xs:namedGroup](#) complexType [214] in [XMLSchema.xsd](#); see [XML source](#) [131]

Component Diagram



XML Representation Summary

```

<xs:sequence
  id = xs:ID
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:element | xs:group | xs:choice | xs:sequence | xs:any)*
</xs:sequence>
    
```

Content model elements (6):

- [xs:annotation](#) [17], [xs:element](#) (type [xs:localElement](#)) [57],
- [xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
- [xs:choice](#) [36], [xs:sequence](#) [128]

Included in content model of elements (1):

- [xs:group](#) [77]

XML Source (see within schema source: p. 406)

```
<xs:element name="sequence" type="xs:simpleExplicitGroup"/>
```

Attribute Detail (all declarations; 2/2)

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [234] within [xs:simpleExplicitGroup](#) complexType; see [XML source](#) [407]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) [234] within [xs:simpleExplicitGroup](#) complexType; see [XML source](#) [407]

↔ [xs:any](#) [20]

Type: [anonymous](#) complexType ([extension of xs:wildcard](#)) [21], complex content

Defined: [by reference](#) [349] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

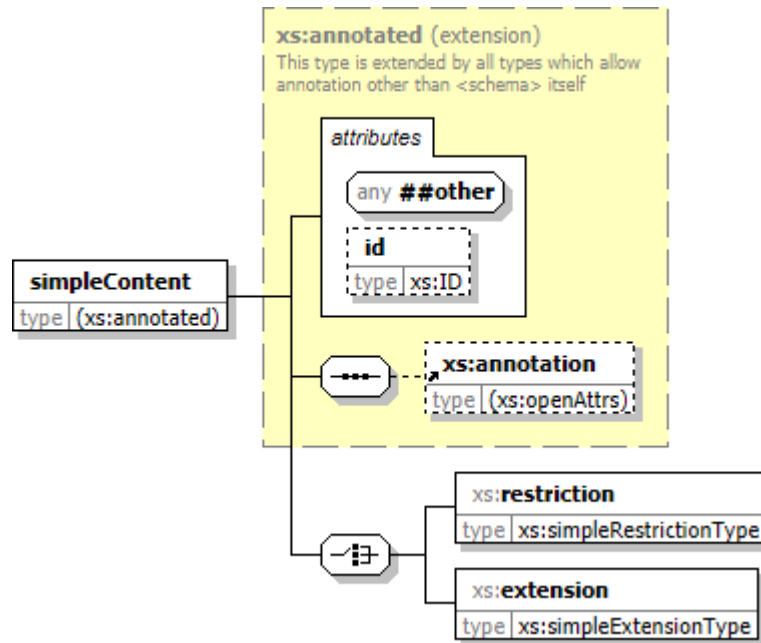
Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

global element

<xs:simpleContent>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: **anonymous** complexType (extension of [xs:annotated](#)) [134]
 Content: complex, 1 attribute, attr. wildcard, 3 elements
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [134]

Component Diagram



XML Representation Summary

```
<xs:simpleContent
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:restriction | xs:extension)
</xs:simpleContent>
```

Content model elements (3):

[xs:annotation](#) [17], [xs:restriction](#) (in [xs:simpleContent](#)) [121],
[xs:extension](#) (in [xs:simpleContent](#)) [70],

Included in content model of elements (2):

[xs:complexType](#) [44], [xs:complexType](#) (type [xs:localComplexType](#)) [48]

Known Usage Locations

- Within model groups (1):
[xs:complexTypeModel](#) [343]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-simpleContent>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType
    
```

XML Source (see within schema source: p. 404)

```

<xs:element id="simpleContent" name="simpleContent">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-simpleContent"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:choice>
          <xs:element name="restriction" type="xs:simpleRestrictionType"/>
          <xs:element name="extension" type="xs:simpleExtensionType"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

Attribute Detail (all declarations; 2/2)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 3/3)

xs:annotation [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

xs:extension [70]

Type: [xs:simpleExtensionType](#) [236], complex content
Defined: [locally](#) within ([this](#)) [xs:simpleContent](#) element; see [XML source](#) [404]

xs:restriction [121]

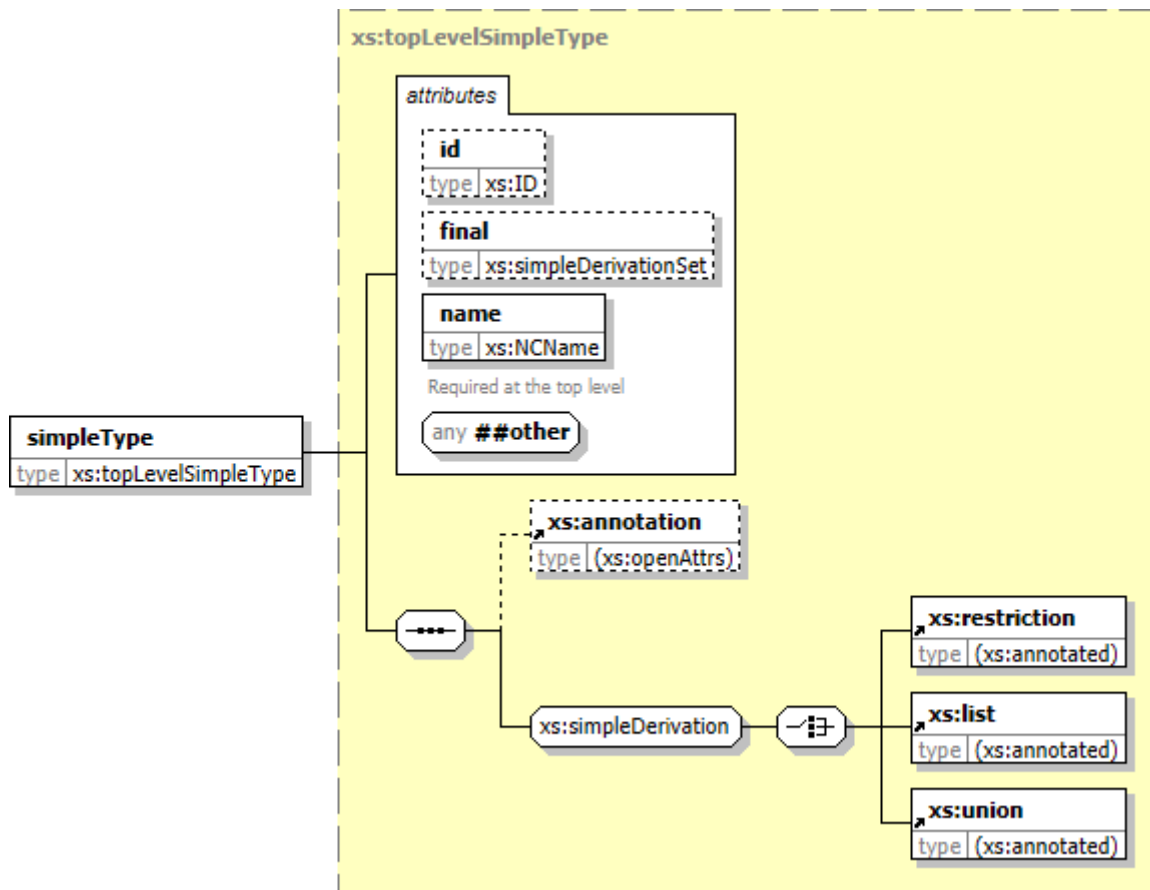
Type: [xs:simpleRestrictionType](#) [239], complex content
Defined: [locally](#) within ([this](#)) [xs:simpleContent](#) element; see [XML source](#) [404]

global element

<xs:simpleType>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:topLevelSimpleType](#) [257]
Content: complex, 3 attributes, attr. wildcard, 4 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [136]

Component Diagram



XML Representation Summary

```
<xs:simpleType
  id = xs:ID
  final = ("#all" | list of ("list" | "union" | "restriction"))
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:restriction | xs:list | xs:union)
</xs:simpleType>
```

Content model elements (4):

[xs:annotation](#) [17], [xs:list](#) [92], [xs:restriction](#) [114], [xs:union](#) [142]

Included in content model of elements (2):

[xs:redefine](#) [111], [xs:schema](#) [7]

Known Usage Locations

- Within model groups (1):

[xs:redefinable](#) [354]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-simpleType>

XML Source (see within schema source: p. 424)

```
<xs:element id="simpleType" name="simpleType" type="xs:topLevelSimpleType">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-simpleType"/>
  </xs:annotation>
</xs:element>
```

Attribute Detail (all declarations; 4/4)

final

Type: [xs:simpleDerivationSet](#) [326]
Use: optional
Defined: [locally](#) [244] within [xs:simpleType](#) complexType; see [XML source](#) [423]

Attribute Value

```
"#all" | list of ("list" | "union" | "restriction")
```

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [258] within [xs:topLevelSimpleType](#) complexType; see [XML source](#) [424]

Required at the top level

{any attribute from non-schema namespace}

Defined: [locally](#) [258] within [xs:topLevelSimpleType](#) complexType; see [XML source](#) [424]

Content Element Detail (all declarations; 4/4)

↔ [xs:annotation](#) [17]


Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [259] within [xs:topLevelSimpleType](#) complexType; see [XML source](#) [423]

↔ [xs:list](#) [92]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [93], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

↔ [xs:restriction](#) [114]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [115], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

 [xs:union](#) [142]

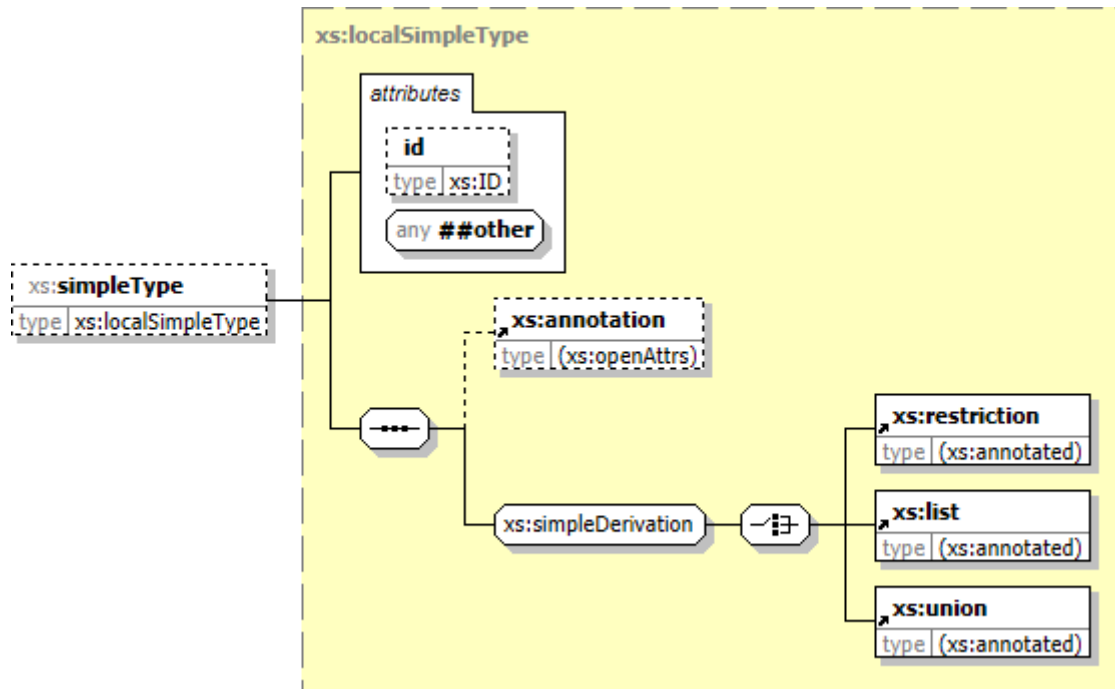
Type: [anonymous complexType](#) ([extension of xs:annotated](#)) [143], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

unified local element

`<xs:simpleType>` (type `xs:localSimpleType`)

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: [xs:localSimpleType](#) [206]
Content: complex, 1 [attribute](#), attr. [wildcard](#), 4 [elements](#)
Block: "#all" (blocks all substitutions of this element or its type)
Defined: locally at 9 [locations](#) in [XMLSchema.xsd](#)

Component Diagram



XML Representation Summary

```

<xs:simpleType
  id = xs:ID
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:restriction | xs:list | xs:union)
</xs:simpleType>
    
```

Content model elements (4):

[xs:annotation](#) [17], [xs:list](#) [92], [xs:restriction](#) [114], [xs:union](#) [142]

Included in content model of elements (9):

[xs:attribute](#) [27], [xs:list](#) [92],
[xs:attribute](#) (type [xs:attribute](#)) [29], [xs:restriction](#) [114],
[xs:element](#) [53], [xs:restriction](#) (in [xs:simpleContent](#)) [121],
[xs:element](#) (type [xs:localElement](#)) [57], [xs:union](#) [142]
[xs:element](#) (type [xs:narrowMaxMin](#)) [61],

Definition Locations

- Within global complexTypes (6):

[xs:attribute](#) [161], [xs:element](#) [178], [xs:localElement](#) [205], [xs:narrowMaxMin](#) [219], [xs:topLevelAttribute](#) [248],
[xs:topLevelElement](#) [256]

- **Within anonymous complexTypes of elements (2):**

[xs:list](#) [93], [xs:union](#) [144]

- **Within model groups (1):**

[xs:simpleRestrictionModel](#) [360]

Annotations (1) (by all definition locations)

Locations (9):

within [xs:attribute](#) complexType [161], within [xs:topLevelAttribute](#) complexType [248], within [xs:element](#) complexType [178], within [xs:topLevelElement](#) complexType [256], within [xs:localElement](#) complexType [205], within [xs:narrowMaxMin](#) complexType [219], within [xs:simpleRestrictionModel](#) group [360], within [xs:list](#) element [93], within [xs:union](#) element [144]

Annotation:

Attribute Detail (all declarations; 2/2)

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [207] within [xs:localSimpleType](#) complexType; see [XML source](#) [424]

Content Element Detail (all declarations; 4/4)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) [207] within [xs:localSimpleType](#) complexType; see [XML source](#) [424]

↔ [xs:list](#) [92]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [93], complex content

Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

↔ [xs:restriction](#) [114]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [115], complex content

Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

↔ [xs:union](#) [142]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [143], complex content

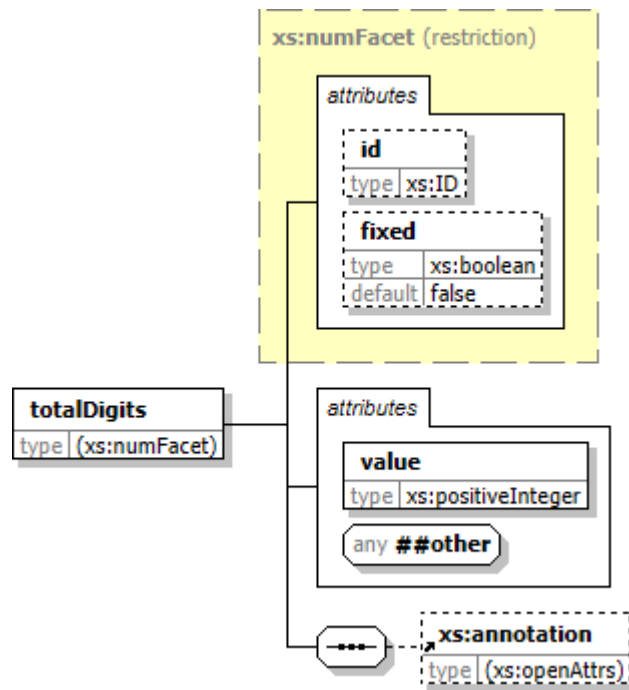
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

global element

<xs:totalDigits>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (restriction of xs:numFacet) [141]
Content: complex, 3 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [141]

Component Diagram



XML Representation Summary

```

<xs:totalDigits
  id      = xs:ID
  fixed  = xs:boolean : "false"
  value  = xs:positiveInteger
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:totalDigits>
    
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-totalDigits>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   ├── xs:annotated [153] (extension)
│   │   ├── xs:facet [186] (restriction)
│   │   │   └── xs:numFacet [222] (restriction)
│   │   └── complexType
│   └──
└──

```

XML Source (see within schema source: p. 426)

```

<xs:element id="totalDigits" name="totalDigits">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-totalDigits"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:numFacet">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="value" type="xs:positiveInteger" use="required"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 4/4)

fixed

Type: `xs:boolean` [270]
Use: optional
Defined: locally [187] within `xs:facet` complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: `xs:ID` [295]
Use: optional
Defined: locally [155] within `xs:annotated` complexType; see [XML source](#) [397]

value

Type: `xs:positiveInteger` [320]
Use: required
Defined: locally within (this) `xs:totalDigits` element; see [XML source](#) [426]

{any attribute from non-schema namespace}

Defined: locally within (this) `xs:totalDigits` element; see [XML source](#) [426]

Content Element Detail (all declarations; 1/1)

`xs:annotation` [17]

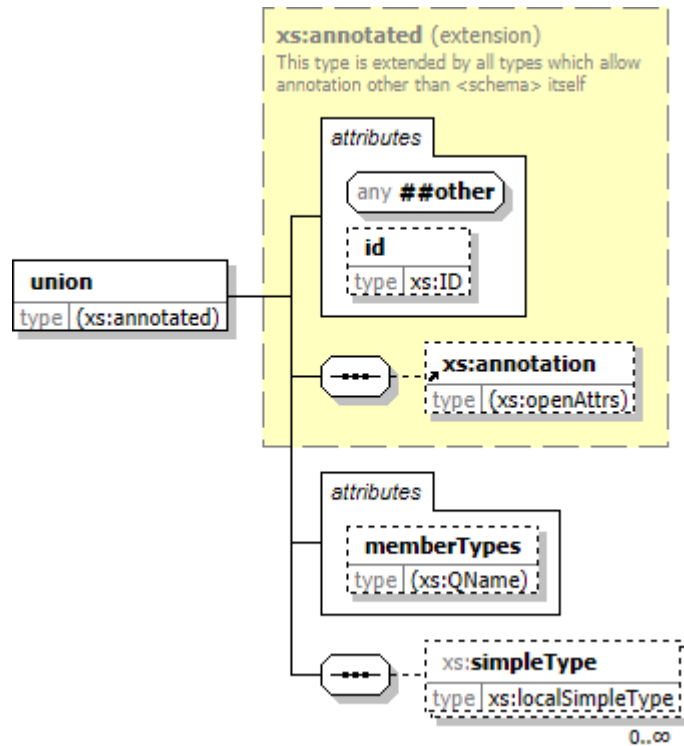
Type: anonymous complexType (extension of `xs:openAttrs`) [18], complex content
Defined: by reference within (this) `xs:totalDigits` element; see [XML source](#) [426]

global element

<xs:union>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (extension of [xs:annotated](#)) [143]
Content: complex, 2 attributes, attr. wildcard, 2 elements
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [143]

Component Diagram



XML Representation Summary

```
<xs:union
  id           = xs:ID
  memberTypes = list of xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType*
</xs:union>
```

Content model elements (2):

[xs:annotation](#) [17], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Included in content model of elements (2):

[xs:simpleType](#) [135], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Known Usage Locations

- Within model groups (1):
[xs:simpleDerivation](#) [357]

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── complexType

```

Annotation

memberTypes attribute must be non-empty or there must be at least one simpleType child

See: <http://www.w3.org/TR/xmlschema-2/#element-union>

XML Source (see within schema source: p. 425)

```

<xs:element id="union" name="union">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-union">
        memberTypes attribute must be non-empty or there must be
        at least one simpleType child
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
        </xs:sequence>
        <xs:attribute name="memberTypes" use="optional">
          <xs:simpleType>
            <xs:list itemType="xs:QName"/>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

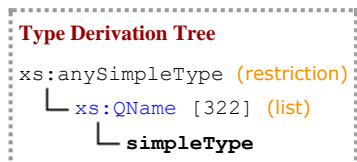
memberTypes

Type: [anonymous simpleType](#) ([list of xs:QName](#)) [144]
Use: optional
Defined: [locally](#) within ([this](#)) [xs:union](#) element; see [XML source](#) [425]

Attribute Value

```
list of xs:QName
```

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 2/2)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:simpleType](#) [138]

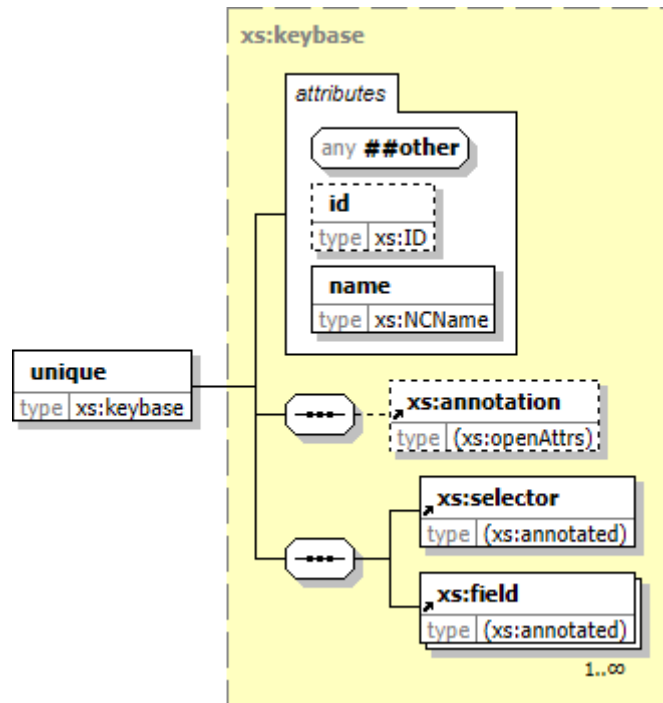
Type: [xs:localSimpleType](#) [206], complex content
Defined: locally within ([this](#)) [xs:union](#) element; see [XML source](#) [425]

global element

<xs:unique>

Namespace: <http://www.w3.org/2001/XMLSchema>
 Type: [xs:keybase](#) [195]
 Content: complex, 2 attributes, attr. wildcard, 3 elements
 Block: "#all" (blocks all substitutions of this element or its type)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [146]

Component Diagram



XML Representation Summary

```
<xs:unique
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:selector, xs:field+
</xs:unique>
```

Content model elements (3):

[xs:annotation](#) [17], [xs:field](#) [72], [xs:selector](#) [125]

Included in content model of elements (3):

[xs:element](#) [53], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]
[xs:element](#) (type [xs:localElement](#)) [57],

Known Usage Locations

- Within model groups (1):
[xs:identityConstraint](#) [348]

Annotation

See: <http://www.w3.org/TR/xmlschema-1/#element-unique>

XML Source (see within schema source: p. 412)

```
<xs:element id="unique" name="unique" type="xs:keybase">  
  <xs:annotation>  
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-unique"/>  
  </xs:annotation>  
</xs:element>
```

Attribute Detail (all declarations; 3/3)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

{any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 3/3)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:field](#) [72]

Type: [anonymous](#) complexType ([extension of](#) [xs:annotated](#)) [73], complex content
Defined: [by reference](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

↔ [xs:selector](#) [125]

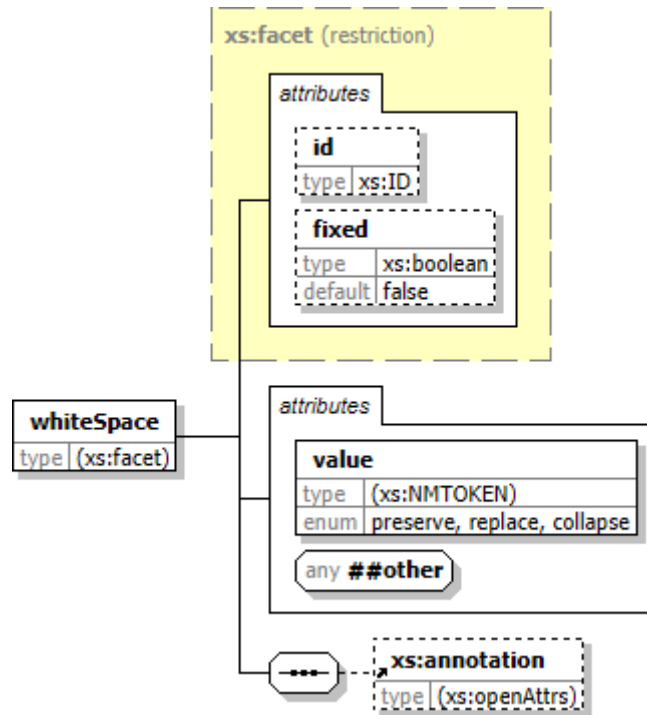
Type: [anonymous](#) complexType ([extension of](#) [xs:annotated](#)) [126], complex content
Defined: [by reference](#) [196] within [xs:keybase](#) complexType; see [XML source](#) [411]

global element

<xs:whiteSpace>

Namespace: <http://www.w3.org/2001/XMLSchema>
Type: anonymous complexType (restriction of `xs:facet`) [148]
Content: complex, 3 attributes, attr. wildcard, 1 element
Block: "#all" (blocks all substitutions of this element or its type)
Defined: globally in `XMLSchema.xsd`; see [XML source](#) [148]

Component Diagram



XML Representation Summary

```
<xs:whiteSpace
  id = xs:ID
  fixed = xs:boolean : "false"
  value = ("preserve" | "replace" | "collapse")
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</xs:whiteSpace>
```

Content model elements (1):

[xs:annotation](#) [17]

Included in content model of elements (2):

[xs:restriction](#) [114], [xs:restriction](#) (in `xs:simpleContent`) [121]

Known Usage Locations

- Within model groups (1):

[xs:facets](#) [346]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#element-whiteSpace>

Anonymous Type Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   ├── xs:annotated [153] (extension)
│   │   ├── xs:facet [186] (restriction)
│   │   └── complexType
│   └──
└──

```

XML Source (see within schema source: p. 426)

```

<xs:element id="whiteSpace" name="whiteSpace">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-whiteSpace"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:facet">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="value" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="preserve"/>
              <xs:enumeration value="replace"/>
              <xs:enumeration value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Attribute Detail (all declarations; 4/4)

fixed

Type: `xs:boolean` [270]
Use: optional
Defined: locally [187] within `xs:facet` complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: `xs:ID` [295]
Use: optional
Defined: locally [155] within `xs:annotated` complexType; see [XML source](#) [397]

value

Type: anonymous simpleType (restriction of `xs:NMTOKEN`) [149]
Use: required
Defined: locally within (this) `xs:whiteSpace` element; see [XML source](#) [426]

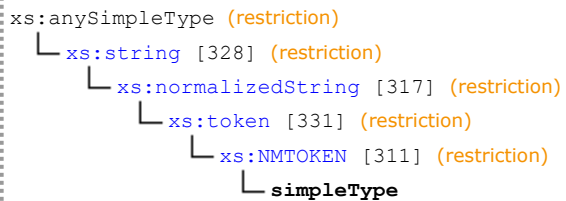
Attribute Value

enumeration of [xs:NMTOKEN](#)

Enumeration: "preserve", "replace", "collapse"

Anonymous simpleType

Type Derivation Tree



■ {any attribute from non-schema namespace}

Defined: locally within (this) [xs:whiteSpace](#) element; see [XML source](#) [427]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content

Defined: by reference within (this) [xs:whiteSpace](#) element; see [XML source](#) [426]

Complex Types

complexType

xs:all

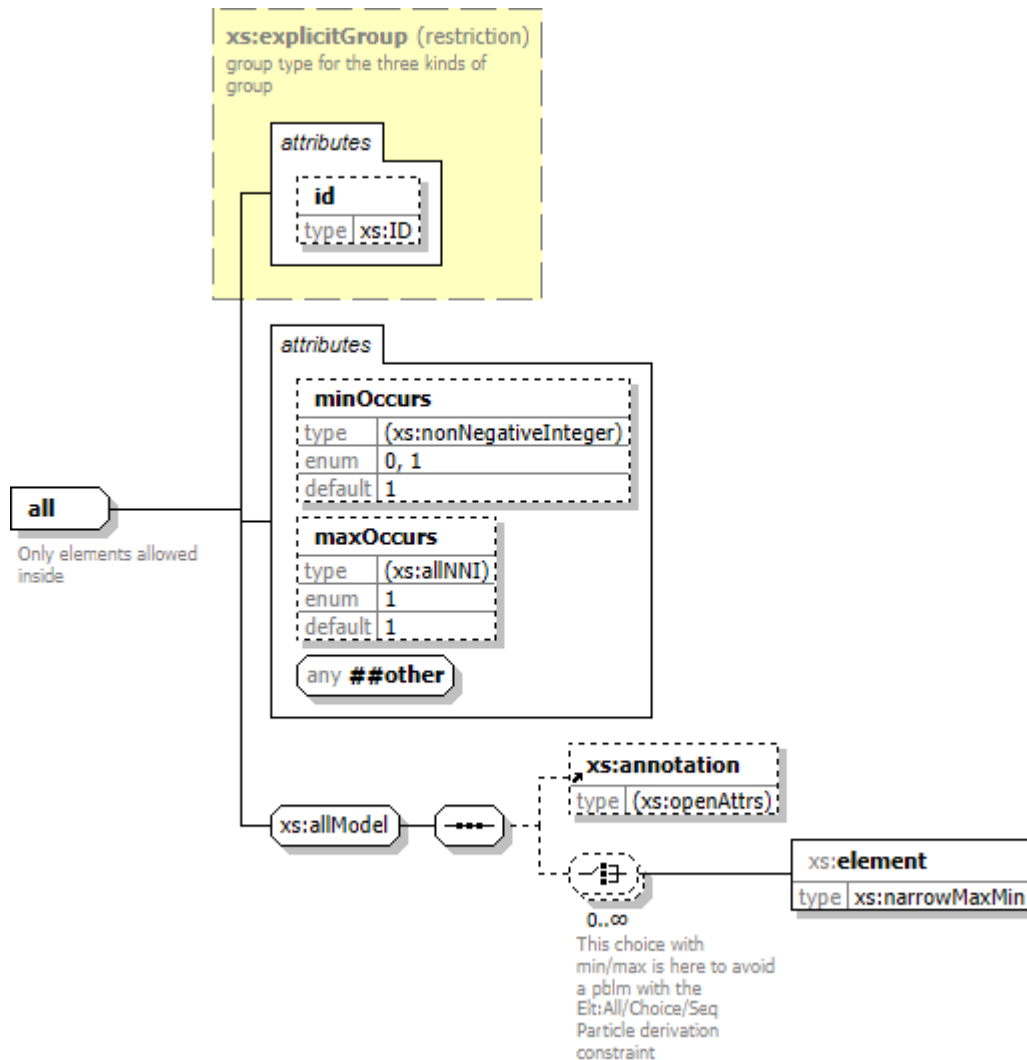
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 3 [attributes](#), attr. [wildcard](#), 2 [elements](#)

Block: "#all" (blocks all substitutions of this complex type through *xsi:type* attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [151]

Component Diagram



XML Representation Summary

```

<...
  <id type="xs:ID"/>
  <minOccurs type="xs:nonNegativeInteger" enum="0, 1" default="1"/>
  <maxOccurs type="xs:allNNI" enum="1" default="1"/>
  {any attribute from non-schema namespace}
  >
  Content: xs:annotation?, xs:element*
</...>
    
```

Content Model Elements (2):

[xs:annotation](#) [17], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

All Direct / Indirect Based Elements (2):

[xs:all](#) [12], [xs:all](#) (in [xs:group](#)) [15]

Known Usage Locations

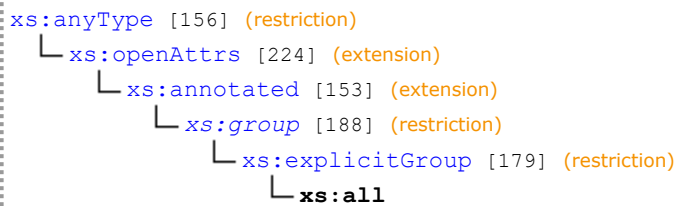
- **As direct type of elements (1):**
[xs:all](#) [12]
- **In derivations of anonymous types of elements (1):**
[xs:all](#) (in [xs:group](#)) [15] (as restriction base)

Annotation

Only elements allowed inside

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 407)

```

<xs:complexType name="all">
  <xs:annotation>
    <xs:documentation>
      Only elements allowed inside
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:restriction base="xs:explicitGroup">
      <xs:group ref="xs:allModel"/>
      <xs:attribute default="1" name="minOccurs" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:nonNegativeInteger">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute default="1" name="maxOccurs" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:allNN1">
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

■ id

Type: [xs:ID](#) [295]
Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs

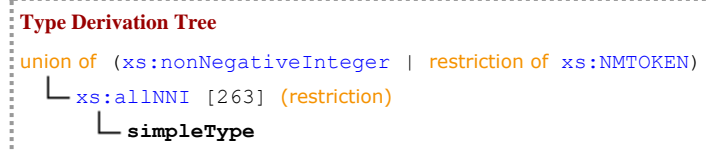
Type: [anonymous](#) simpleType ([restriction of xs:allNNI](#)) [152]
 Use: optional
 Defined: locally within ([this](#)) [xs:all](#) complexType; see [XML source](#) [408]

Attribute Value

enumeration of ([xs:nonNegativeInteger](#) | "unbounded")

Enumeration: "1"
 Default: "1"

Anonymous simpleType



■ minOccurs

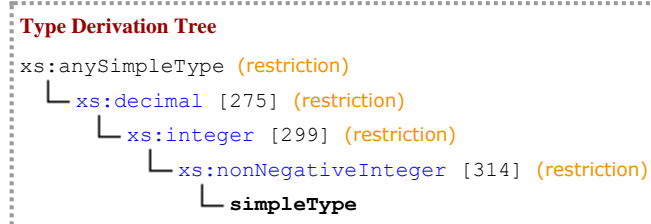
Type: [anonymous](#) simpleType ([restriction of xs:nonNegativeInteger](#)) [152]
 Use: optional
 Defined: locally within ([this](#)) [xs:all](#) complexType; see [XML source](#) [407]

Attribute Value

enumeration of [xs:nonNegativeInteger](#)

Enumeration: "0", "1"
 Default: "1"

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:all](#) complexType; see [XML source](#) [408]

Content Element Detail (all declarations; 2/2)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
 Defined: [by reference](#) [339] within [xs:allModel](#) group; see [XML source](#) [407]

↔ [xs:element](#) [61]

Type: [xs:narrowMaxMin](#) [215], complex content
 Defined: [locally](#) [339] within [xs:allModel](#) group; see [XML source](#) [407]

complexType xs:annotated

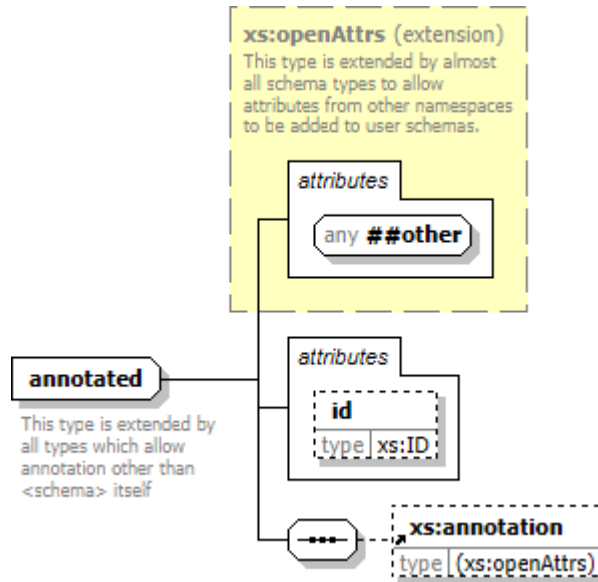
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 1 [attribute](#), attr. [wildcard](#), 1 [element](#)

Block: "#all" (*blocks all substitutions of this complex type through xsi:type attribute in instance XML documents*)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [154]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>
```

Content Model Elements (1):

[xs:annotation](#) [17]

Known Direct Subtypes (11):

[xs:attribute](#) [158], [xs:attributeGroup](#) [162], [xs:complexType](#) [170], [xs:element](#) [174], [xs:extensionType](#) [183], [xs:facet](#) [186], [xs:group](#) [188], [xs:keybase](#) [195], [xs:restrictionType](#) [229], [xs:simpleType](#) [243], [xs:wildcard](#) [260]

Known Indirect Subtypes (21):

[xs:all](#) [150], [xs:attributeGroupRef](#) [165], [xs:complexRestrictionType](#) [167], [xs:explicitGroup](#) [179], [xs:groupRef](#) [192], [xs:localComplexType](#) [197], [xs:localElement](#) [201], [xs:localSimpleType](#) [206], [xs:namedAttributeGroup](#) [209], [xs:namedGroup](#) [212], [xs:narrowMaxMin](#) [215], [xs:noFixedFacet](#) [220], [xs:numFacet](#) [222], [xs:realGroup](#) [226], [xs:simpleExplicitGroup](#) [233], [xs:simpleExtensionType](#) [236], [xs:simpleRestrictionType](#) [239], [xs:topLevelAttribute](#) [246], [xs:topLevelComplexType](#) [249], [xs:topLevelElement](#) [253], [xs:topLevelSimpleType](#) [257]

All Direct / Indirect Based Elements (50):

xs:all [12],	xs:key [85],
xs:all (in xs:group) [15],	xs:keyref [87],
xs:any [20],	xs:length [90],
xs:anyAttribute [23],	xs:list [92],
xs:attribute [27],	xs:maxExclusive [94],

[xs:attribute](#) (type [xs:attribute](#)) [29],
[xs:attributeGroup](#) [32],
[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34],
[xs:choice](#) [36],
[xs:choice](#) (in [xs:group](#)) [39],
[xs:complexContent](#) [41],
[xs:complexType](#) [44],
[xs:complexType](#) (type [xs:localComplexType](#)) [48],
[xs:element](#) [53],
[xs:element](#) (type [xs:localElement](#)) [57],
[xs:element](#) (type [xs:narrowMaxMin](#)) [61],
[xs:enumeration](#) [65],
[xs:extension](#) (in [xs:complexContent](#)) [67],
[xs:extension](#) (in [xs:simpleContent](#)) [70],
[xs:field](#) [72],
[xs:fractionDigits](#) [75],
[xs:group](#) [77],
[xs:group](#) (type [xs:groupRef](#)) [79],
[xs:import](#) [81],
[xs:include](#) [83],
[xs:maxInclusive](#) [96],
[xs:maxLength](#) [98],
[xs:minExclusive](#) [100],
[xs:minInclusive](#) [102],
[xs:minLength](#) [104],
[xs:notation](#) [106],
[xs:pattern](#) [109],
[xs:restriction](#) [114],
[xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:restriction](#) (in [xs:simpleContent](#)) [121],
[xs:selector](#) [125],
[xs:sequence](#) [128],
[xs:sequence](#) (in [xs:group](#)) [131],
[xs:simpleContent](#) [133],
[xs:simpleType](#) [135],
[xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
[xs:totalDigits](#) [140],
[xs:union](#) [142],
[xs:unique](#) [145],
[xs:whiteSpace](#) [147]

Known Usage Locations

- In derivations of other global types (11):**

[xs:attribute](#) [158] (as extension base),
[xs:attributeGroup](#) [162] (as extension base),
[xs:complexType](#) [170] (as extension base),
[xs:element](#) [174] (as extension base),
[xs:extensionType](#) [183] (as extension base),
[xs:facet](#) [186] (as extension base),
[xs:group](#) [188] (as extension base),
[xs:keybase](#) [195] (as extension base),
[xs:restrictionType](#) [229] (as extension base),
[xs:simpleType](#) [243] (as extension base),
[xs:wildcard](#) [260] (as extension base)

- In derivations of anonymous types of elements (10):**

[xs:complexContent](#) [41] (as extension base),
[xs:field](#) [72] (as extension base),
[xs:import](#) [81] (as extension base),
[xs:include](#) [83] (as extension base),
[xs:list](#) [92] (as extension base),
[xs:notation](#) [106] (as extension base),
[xs:restriction](#) [114] (as extension base),
[xs:selector](#) [125] (as extension base),
[xs:simpleContent](#) [133] (as extension base),
[xs:union](#) [142] (as extension base)

Annotation

This type is extended by all types which allow annotation other than <schema> itself

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated
    
```

XML Source (see within schema source: p. 397)

```

<xs:complexType name="annotated">
  <xs:annotation>
    <xs:documentation>
      This type is extended by all types which allow annotation
      other than &lt;schema&gt; itself
    </xs:documentation>
  </xs:annotation>
  
```

```
<xs:complexContent>
  <xs:extension base="xs:openAttrs">
    <xs:sequence>
      <xs:element minOccurs="0" ref="xs:annotation"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 2/2)

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: locally within [\(this\) xs:annotated](#) complexType; see [XML source](#) [397]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content

Defined: by reference within [\(this\) xs:annotated](#) complexType; see [XML source](#) [397]

complexType xs:anyType

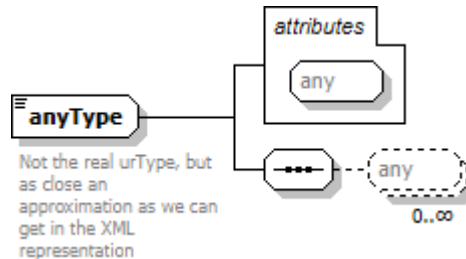
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: mixed (allows character data), attr. [wildcard](#), elem. [wildcard](#)

Block: "#all" (blocks all substitutions of this complex type through *xsi:type* attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [157]

Component Diagram



XML Representation Summary

```
<...
  {any attribute from any namespace}
>
Content: {text} × {any}*
</...>
```

Known Direct Subtypes (1):

[xs:openAttrs](#) [224]

Known Indirect Subtypes (33):

[xs:all](#) [150], [xs:annotated](#) [153], [xs:attribute](#) [158], [xs:attributeGroup](#) [162], [xs:attributeGroupRef](#) [165], [xs:complexRestrictionType](#) [167], [xs:complexType](#) [170], [xs:element](#) [174], [xs:explicitGroup](#) [179], [xs:extensionType](#) [183], [xs:facet](#) [186], [xs:group](#) [188], [xs:groupRef](#) [192], [xs:keybase](#) [195], [xs:localComplexType](#) [197], [xs:localElement](#) [201], [xs:localSimpleType](#) [206], [xs:namedAttributeGroup](#) [209], [xs:namedGroup](#) [212], [xs:narrowMaxMin](#) [215], [xs:noFixedFacet](#) [220], [xs:numFacet](#) [222], [xs:realGroup](#) [226], [xs:restrictionType](#) [229], [xs:simpleExplicitGroup](#) [233], [xs:simpleExtensionType](#) [236], [xs:simpleRestrictionType](#) [239], [xs:simpleType](#) [243], [xs:topLevelAttribute](#) [246], [xs:topLevelComplexType](#) [249], [xs:topLevelElement](#) [253], [xs:topLevelSimpleType](#) [257], [xs:wildcard](#) [260]

All Direct / Indirect Based Elements (53):

[xs:all](#) [12], [xs:all](#) (in [xs:group](#)) [15], [xs:annotation](#) [17], [xs:any](#) [20], [xs:anyAttribute](#) [23], [xs:attribute](#) [27], [xs:attribute](#) (type [xs:attribute](#)) [29], [xs:attributeGroup](#) [32], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34], [xs:choice](#) [36], [xs:choice](#) (in [xs:group](#)) [39], [xs:complexContent](#) [41], [xs:complexType](#) [44], [xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:element](#) [53], [xs:element](#) (type [xs:localElement](#)) [57], [xs:element](#) (type [xs:narrowMaxMin](#)) [61], [xs:enumeration](#) [65], [xs:extension](#) (in [xs:complexContent](#)) [67], [xs:keyref](#) [87], [xs:length](#) [90], [xs:list](#) [92], [xs:maxExclusive](#) [94], [xs:maxInclusive](#) [96], [xs:maxLength](#) [98], [xs:minExclusive](#) [100], [xs:minInclusive](#) [102], [xs:minLength](#) [104], [xs:notation](#) [106], [xs:pattern](#) [109], [xs:redefine](#) [111], [xs:restriction](#) [114], [xs:restriction](#) (in [xs:complexContent](#)) [118], [xs:restriction](#) (in [xs:simpleContent](#)) [121], [xs:schema](#) [7], [xs:selector](#) [125], [xs:sequence](#) [128], [xs:sequence](#) (in [xs:group](#)) [131],

[xs:extension](#) (in [xs:simpleContent](#)) [70],
[xs:field](#) [72],
[xs:fractionDigits](#) [75],
[xs:group](#) [77],
[xs:group](#) (type [xs:groupRef](#)) [79],
[xs:import](#) [81],
[xs:include](#) [83],
[xs:key](#) [85],

[xs:simpleContent](#) [133],
[xs:simpleType](#) [135],
[xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
[xs:totalDigits](#) [140],
[xs:union](#) [142],
[xs:unique](#) [145],
[xs:whiteSpace](#) [147]

Known Usage Locations

- In derivations of other global types (1):

[xs:openAttrs](#) [224] (as restriction base)

Annotation

Not the real urType, but as close an approximation as we can get in the XML representation

XML Source (see within schema source: p. 413)

```
<xs:complexType mixed="true" name="anyType">
  <xs:annotation>
    <xs:documentation>
      Not the real urType, but as close an approximation as we can
      get in the XML representation
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any maxOccurs="unbounded" minOccurs="0" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
```

Attribute Detail (all declarations; 1/1)

■ *{any attribute from any namespace}*

Defined: locally within (this) [xs:anyType](#) complexType; see [XML source](#) [413]

Content Element Detail (all declarations; 1/1)

↔ *{any element from any namespace}*

Defined: locally within (this) [xs:anyType](#) complexType; see [XML source](#) [413]

complexType xs:attribute

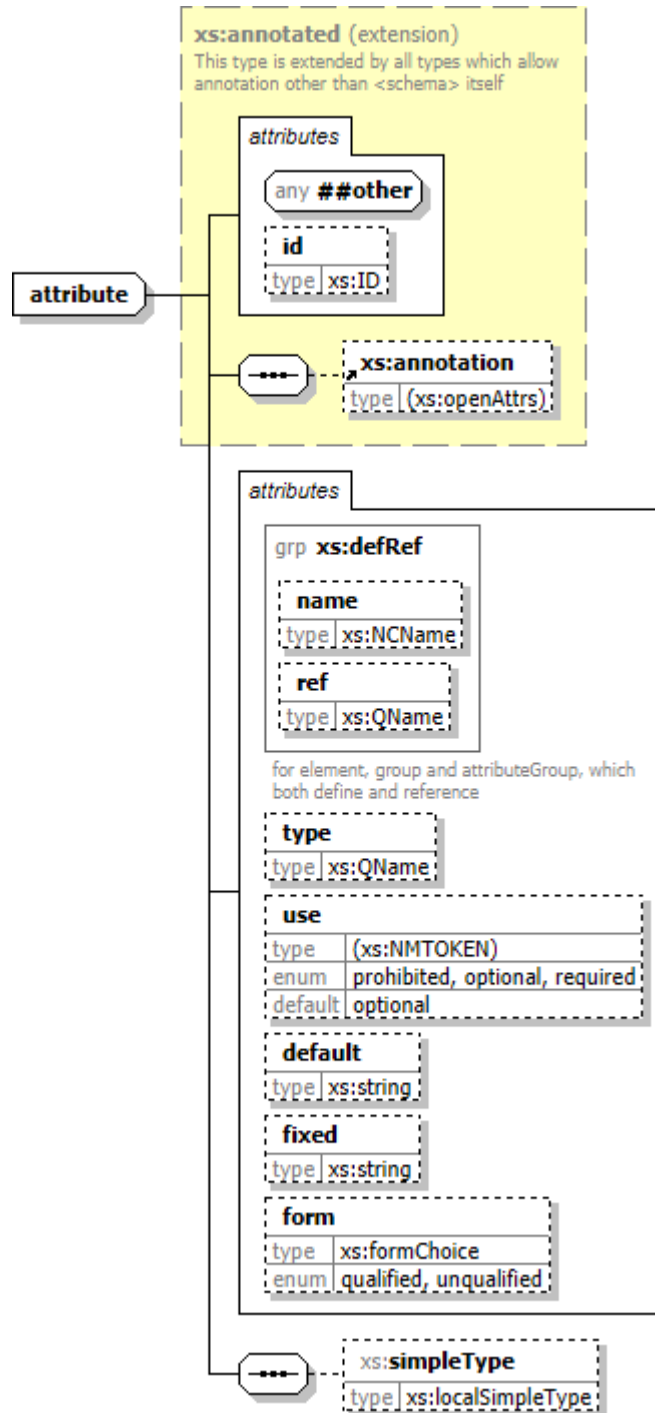
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 8 attributes, attr. wildcard, 2 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [159]

Component Diagram



XML Representation Summary

```
<...
  id       = xs:ID
  name     = xs:NCName
  ref      = xs:QName
  type     = xs:QName
  use     = ("prohibited" | "optional" | "required") : "optional"
  default = xs:string
  fixed   = xs:string
  form    = ("qualified" | "unqualified")
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType?
</...>
```

Content Model Elements (2):

[xs:annotation](#) [17], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Known Direct Subtypes (1):

[xs:topLevelAttribute](#) [246]

All Direct / Indirect Based Elements (2):

[xs:attribute](#) [27], [xs:attribute](#) (type [xs:attribute](#)) [29]

Known Usage Locations

- In derivations of other global types (1):
 - [xs:topLevelAttribute](#) [246] (as restriction base)
- As direct type of elements (1):
 - [xs:attribute](#) (type [xs:attribute](#)) [29]

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:attribute
```

XML Source (see within schema source: p. 401)

```
<xs:complexType name="attribute">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
      </xs:sequence>
      <xs:attributeGroup ref="xs:defRef"/>
      <xs:attribute name="type" type="xs:QName"/>
      <xs:attribute default="optional" name="use" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="prohibited"/>
            <xs:enumeration value="optional"/>
            <xs:enumeration value="required"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="default" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<xs:attribute name="fixed" type="xs:string"/>
<xs:attribute name="form" type="xs:formChoice"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 9/9)

■ default

Type: [xs:string](#) [328]
Use: optional
Defined: locally within ([this](#)) [xs:attribute](#) complexType; see [XML source](#) [401]

■ fixed

Type: [xs:string](#) [328]
Use: optional
Defined: locally within ([this](#)) [xs:attribute](#) complexType; see [XML source](#) [401]

■ form

Type: [xs:formChoice](#) [286]
Use: optional
Defined: locally within ([this](#)) [xs:attribute](#) complexType; see [XML source](#) [401]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ name

Type: [xs:NCName](#) [308]
Use: optional
Defined: locally [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ ref

Type: [xs:QName](#) [322]
Use: optional
Defined: locally [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: locally within ([this](#)) [xs:attribute](#) complexType; see [XML source](#) [401]

■ use

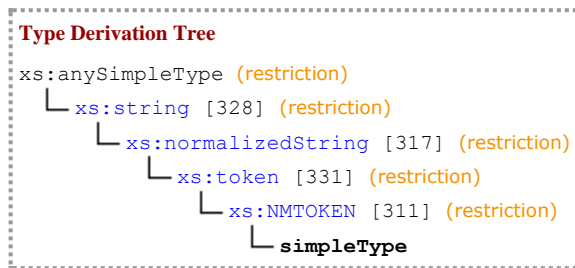
Type: [anonymous](#) simpleType ([restriction of \[xs:NMTOKEN\]\(#\)](#)) [161]
Use: optional
Defined: locally within ([this](#)) [xs:attribute](#) complexType; see [XML source](#) [401]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "prohibited", "optional", "required"
Default: "optional"

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 2/2)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: locally within ([this](#)) [xs:attribute](#) complexType; see [XML source](#) [401]

complexType xs:attributeGroup

Namespace: <http://www.w3.org/2001/XMLSchema>

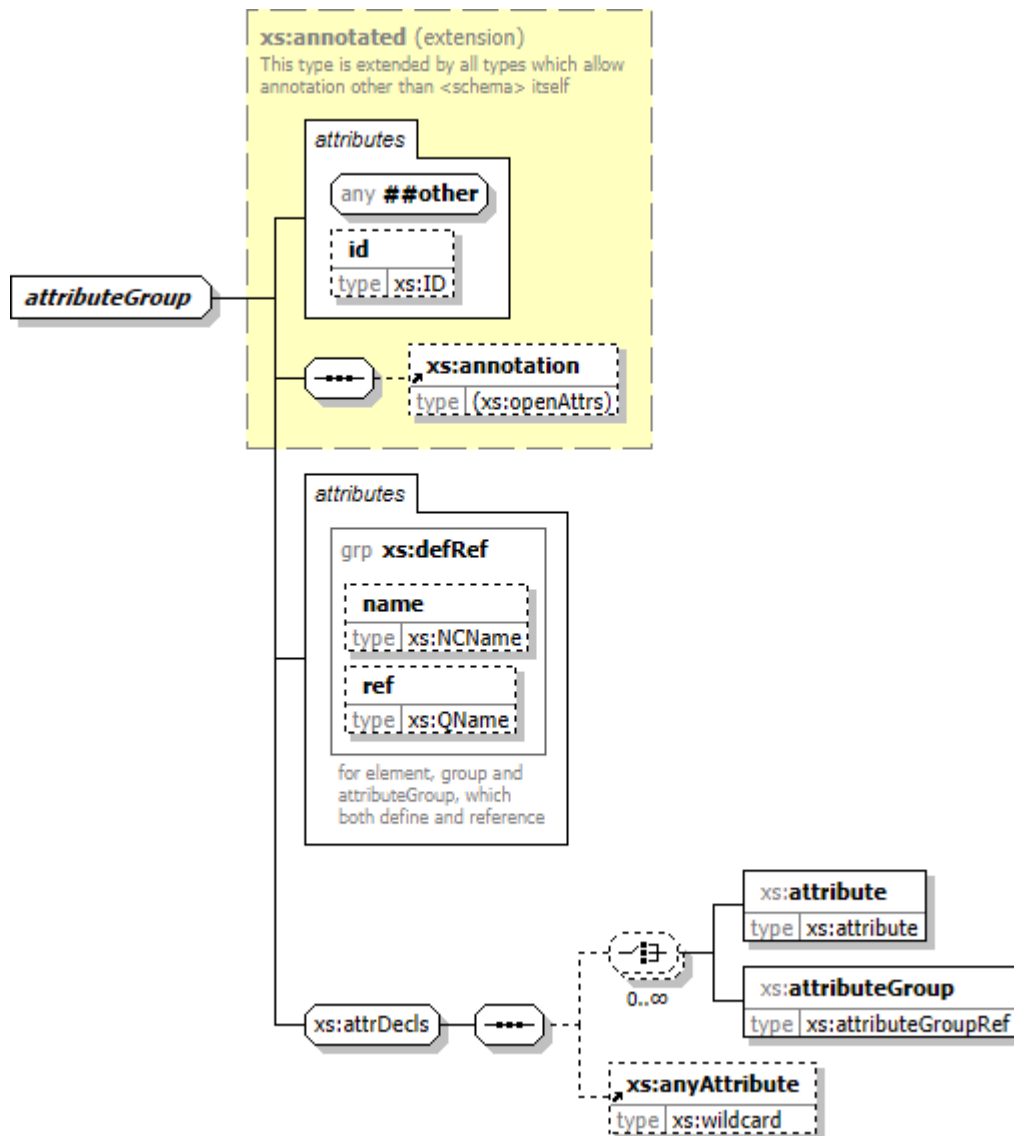
Content: complex, 3 attributes, attr. wildcard, 4 elements

Abstract: (cannot be assigned directly to elements used in instance XML documents)

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [163]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  name = xs:NCName
  ref = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?
</...>

```

Content Model Elements (4):

[xs:annotation](#) [17], [xs:attribute](#) (type [xs:attribute](#)) [29],

[xs:anyAttribute](#) [23], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

Known Direct Subtypes (2):

[xs:attributeGroupRef](#) [165], [xs:namedAttributeGroup](#) [209]

All Direct / Indirect Based Elements (2):

[xs:attributeGroup](#) [32], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

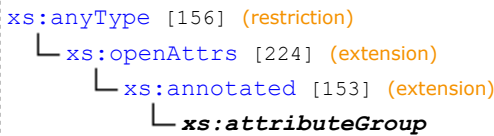
Known Usage Locations

- In derivations of other global types (2):

[xs:attributeGroupRef](#) [165] (as restriction base), [xs:namedAttributeGroup](#) [209] (as restriction base)

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 409)

```
<xs:complexType abstract="true" name="attributeGroup">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group ref="xs:attrDecls"/>
      <xs:attributeGroup ref="xs:defRef"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 4/4)

- id
Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]
- name
Type: [xs:NCName](#) [308]
Use: optional
Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]
- ref
Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]
- {any attribute from non-schema namespace}
Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 4/4)

[xs:annotation](#) [17]

Type: [anonymous complexType](#) ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

complexType xs:attributeGroupRef

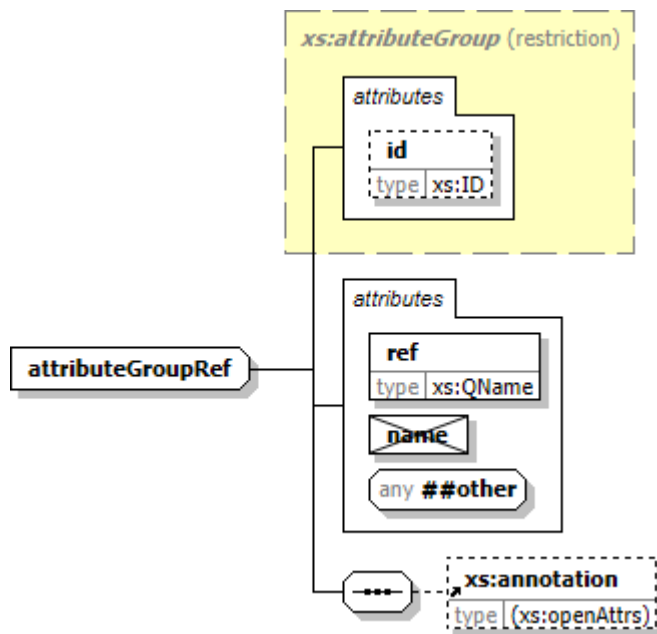
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 1 element

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [166]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  ref = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>
```

Content Model Elements (1):

[xs:annotation](#) [17]

All Direct / Indirect Based Elements (1):

[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

Known Usage Locations

- As direct type of elements (1):

[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:attributeGroup [162] (restriction)
│           └── xs:attributeGroupRef
    
```

XML Source (see within schema source: p. 409)

```

<xs:complexType name="attributeGroupRef">
  <xs:complexContent>
    <xs:restriction base="xs:attributeGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="ref" type="xs:QName" use="required"/>
      <xs:attribute name="name" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

- id
 - Type: [xs:ID](#) [295]
 - Use: optional
 - Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

- name
 - Use: prohibited

- ref
 - Type: [xs:QName](#) [322]
 - Use: required
 - Defined: locally within ([this](#)) [xs:attributeGroupRef](#) complexType; see [XML source](#) [409]

- {any attribute from non-schema namespace}
 - Defined: locally within ([this](#)) [xs:attributeGroupRef](#) complexType; see [XML source](#) [410]

Content Element Detail (all declarations; 1/1)

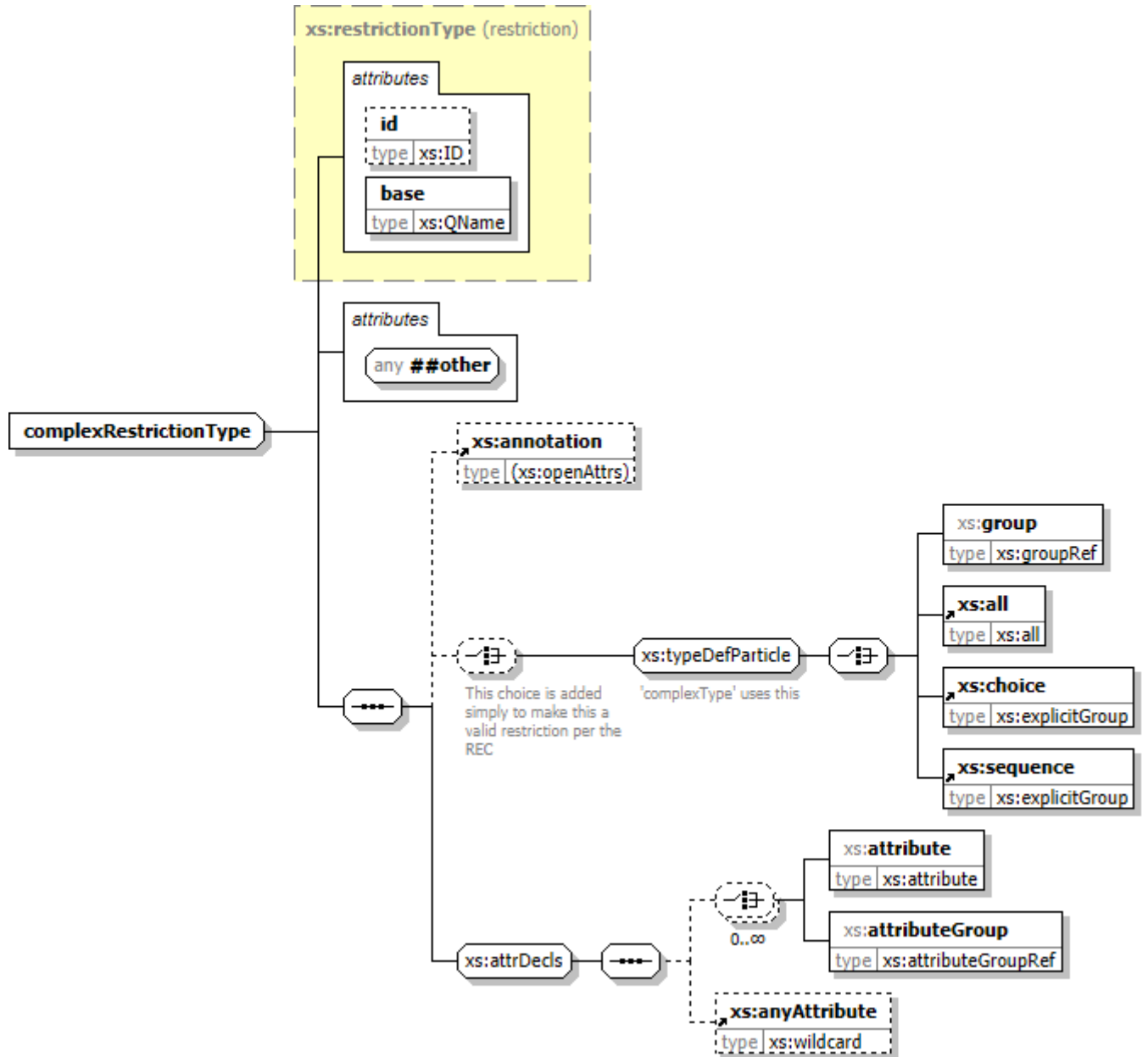
- ↔ [xs:annotation](#) [17]
 - Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
 - Defined: by reference within ([this](#)) [xs:attributeGroupRef](#) complexType; see [XML source](#) [409]

complexType

xs:complexTypeRestrictionType

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 2 attributes, attr. wildcard, 8 elements
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [168]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:group | xs:all | xs:choice | xs:sequence)?, (xs:attribute |
xs:attributeGroup)*, xs:anyAttribute?
</...>

```

Content Model Elements (8):

[xs:all](#) [12], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34],
[xs:annotation](#) [17], [xs:choice](#) [36],
[xs:anyAttribute](#) [23], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:attribute](#) (type [xs:attribute](#)) [29], [xs:sequence](#) [128]

All Direct / Indirect Based Elements (1):

[xs:restriction](#) (in [xs:complexTypeRestrictionType](#)) [118]

Known Usage Locations

- As direct type of elements (1):

[xs:restriction](#) (in [xs:complexTypeRestrictionType](#)) [118]

Type Definition Detail



XML Source (see within schema source: p. 402)

```

<xs:complexType name="complexTypeRestrictionType">
  <xs:complexContent>
    <xs:restriction base="xs:restrictionType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              This choice is added simply to
              make this a valid restriction per the REC
            </xs:documentation>
          </xs:annotation>
          <xs:group ref="xs:typeDefParticle"/>
        </xs:choice>
        <xs:group ref="xs:attrDecls"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 3/3)

■ base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [231] within [xs:restrictionType](#) complexType; see [XML source](#) [402]

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ {any attribute from non-schema namespace}

Defined: locally within (this) [xs:complexRestrictionType](#) complexType; see [XML source](#) [403]

Content Element Detail (all declarations; 8/8)

↔ [xs:all](#) [12]

Type: [xs:all](#) [150], complex content

Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content

Defined: by reference within (this) [xs:complexRestrictionType](#) complexType; see [XML source](#) [402]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content

Defined: by reference [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content

Defined: locally [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content

Defined: locally [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: locally [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

complexType

xs:complexType

Namespace: <http://www.w3.org/2001/XMLSchema>

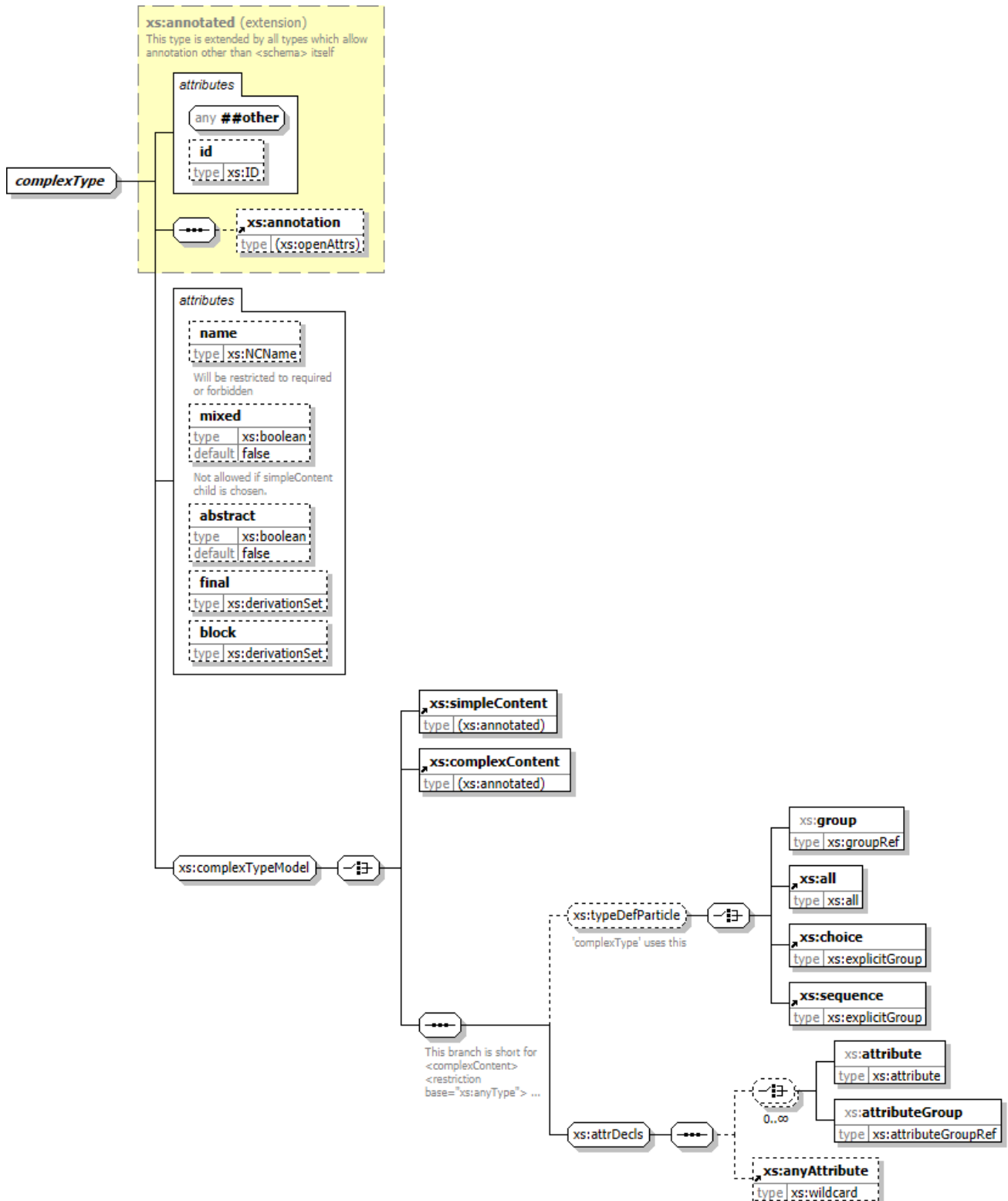
Content: complex, 6 attributes, attr. wildcard, 10 elements

Abstract: (cannot be assigned directly to elements used in instance XML documents)

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [171]

Component Diagram



XML Representation Summary

```
<...
  id           = xs:ID
  name        = xs:NCName
  mixed       = xs:boolean : "false"
  abstract    = xs:boolean : "false"
  final      = ("#all" | list of ("extension" | "restriction"))
  block      = ("#all" | list of ("extension" | "restriction"))
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleContent | xs:complexContent | ((xs:group | xs:all | xs:choice |
xs:sequence)?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?))
</...>
```

Content Model Elements (10):

xs:all [12],	xs:choice [36],
xs:annotation [17],	xs:complexContent [41],
xs:anyAttribute [23],	xs:group (type xs:groupRef) [79],
xs:attribute (type xs:attribute) [29],	xs:sequence [128],
xs:attributeGroup (type xs:attributeGroupRef) [34],	xs:simpleContent [133]

Known Direct Subtypes (2):

[xs:localComplexType](#) [197], [xs:topLevelComplexType](#) [249]

All Direct / Indirect Based Elements (2):

[xs:complexType](#) [44], [xs:complexType](#) (type [xs:localComplexType](#)) [48]

Known Usage Locations

- In derivations of other global types (2):

[xs:localComplexType](#) [197] (as restriction base), [xs:topLevelComplexType](#) [249] (as restriction base)

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:complexType
```

XML Source (see within schema source: p. 402)

```
<xs:complexType abstract="true" name="complexType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group ref="xs:complexTypeModel"/>
      <xs:attribute name="name" type="xs:NCName">
        <xs:annotation>
          <xs:documentation>
            Will be restricted to required or forbidden
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute default="false" name="mixed" type="xs:boolean" use="optional">
        <xs:annotation>
          <xs:documentation>
            Not allowed if simpleContent child is chosen.
            May be overridden by setting on complexContent child.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

<xs:attribute default="false" name="abstract" type="xs:boolean" use="optional"/>
<xs:attribute name="final" type="xs:derivationSet"/>
<xs:attribute name="block" type="xs:derivationSet"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Attribute Detail (all declarations; 7/7)

abstract

Type: xs:boolean [270]
Use: optional
Defined: locally within (this) xs:complexType complexType; see XML source [402]

Attribute Value

Default: "false"

block

Type: xs:derivationSet [279]
Use: optional
Defined: locally within (this) xs:complexType complexType; see XML source [402]

Attribute Value

"#all" | list of ("extension" | "restriction")

final

Type: xs:derivationSet [279]
Use: optional
Defined: locally within (this) xs:complexType complexType; see XML source [402]

Attribute Value

"#all" | list of ("extension" | "restriction")

id

Type: xs:ID [295]
Use: optional
Defined: locally [155] within xs:annotated complexType; see XML source [397]

mixed

Type: xs:boolean [270]
Use: optional
Defined: locally within (this) xs:complexType complexType; see XML source [402]

Not allowed if simpleContent child is chosen.
 May be overridden by setting on complexContent child.

Attribute Value

Default: "false"

name

Type: xs:NCName [308]
Use: optional
Defined: locally within (this) xs:complexType complexType; see XML source [402]

Will be restricted to required or forbidden

{any attribute from non-schema namespace}

Defined: locally [225] within xs:openAttrs complexType; see XML source [397]

Content Element Detail (all declarations; 10/10)

[xs:all](#) [12]

Type: [xs:all](#) [150], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:annotation](#) [17]

Type: [anonymous complexType \(extension of xs:openAttrs\)](#) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated complexType](#); see [XML source](#) [397]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:complexContent](#) [41]

Type: [anonymous complexType \(extension of xs:annotated\)](#) [42], complex content
Defined: [by reference](#) [343] within [xs:complexTypeModel](#) group; see [XML source](#) [401]

[xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content
Defined: [locally](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

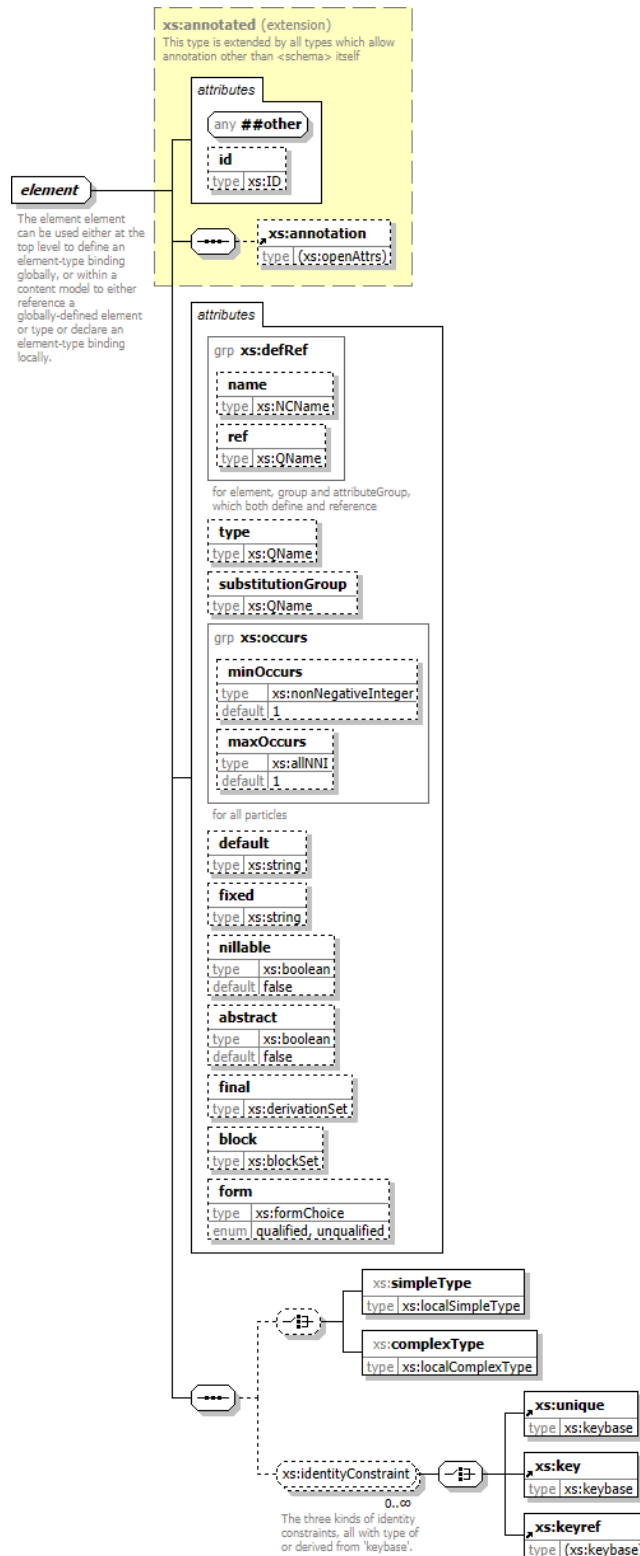
[xs:simpleContent](#) [133]

Type: [anonymous complexType \(extension of xs:annotated\)](#) [134], complex content
Defined: [by reference](#) [343] within [xs:complexTypeModel](#) group; see [XML source](#) [401]

complexType xs:element

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 14 attributes, attr. wildcard, 6 elements
Abstract: (cannot be assigned directly to elements used in instance XML documents)
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [176]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  name = xs:NCName
  ref = xs:QName
  type = xs:QName
  substitutionGroup = xs:QName
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
  default = xs:string
  fixed = xs:string
  nillable = xs:boolean : "false"
  abstract = xs:boolean : "false"
  final = ("#all" | list of ("extension" | "restriction"))
  block = ("#all" | list of ("extension" | "restriction" | "substitution"))
  form = ("qualified" | "unqualified")
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</...>

```

Content Model Elements (6):

[xs:annotation](#) [17], [xs:keyref](#) [87],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
[xs:key](#) [85], [xs:unique](#) [145]

Known Direct Subtypes (2):

[xs:localElement](#) [201], [xs:topLevelElement](#) [253]

Known Indirect Subtypes (1):

[xs:narrowMaxMin](#) [215]

All Direct / Indirect Based Elements (3):

[xs:element](#) [53], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]
[xs:element](#) (type [xs:localElement](#)) [57],

Known Usage Locations

- In derivations of other global types (2):

[xs:localElement](#)[201] (as restriction base), [xs:topLevelElement](#)[253] (as restriction base)

Annotation

The element element can be used either at the top level to define an element-type binding globally, or within a content model to either reference a globally-defined element or type or declare an element-type binding locally. The ref form is not allowed at the top level.

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:element
    
```

XML Source (see within schema source: p. 404)

```

<xs:complexType abstract="true" name="element">
  <xs:annotation>
    <xs:documentation>
      The element element can be used either at the top level to define an element-type binding globally, or within a content model to either reference a globally-defined element or type or declare an element-type binding locally. The ref form is not allowed at the top level.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
      <xs:attributeGroup ref="xs:defRef"/>
      <xs:attribute name="type" type="xs:QName"/>
      <xs:attribute name="substitutionGroup" type="xs:QName"/>
      <xs:attributeGroup ref="xs:occurs"/>
      <xs:attribute name="default" type="xs:string"/>
      <xs:attribute name="fixed" type="xs:string"/>
      <xs:attribute default="false" name="nillable" type="xs:boolean" use="optional"/>
      <xs:attribute default="false" name="abstract" type="xs:boolean" use="optional"/>
      <xs:attribute name="final" type="xs:derivationSet"/>
      <xs:attribute name="block" type="xs:blockSet"/>
      <xs:attribute name="form" type="xs:formChoice"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 15/15)

abstract

Type: `xs:boolean` [270]
Use: optional
Defined: locally within (this) `xs:element` complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

block

Type: `xs:blockSet` [268]
Use: optional
Defined: locally within (this) `xs:element` complexType; see [XML source](#) [405]

Attribute Value

```

"#all" | list of ("extension" | "restriction" | "substitution")
    
```

■ default

Type: [xs:string](#) [328]
Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

■ final

Type: [xs:derivationSet](#) [279]
Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

■ fixed

Type: [xs:string](#) [328]
Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

■ form

Type: [xs:formChoice](#) [286]
Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ maxOccurs

Type: [xs:allNNI](#) [263]
Use: optional
Defined: locally [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

■ minOccurs

Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: locally [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

■ name

Type: [xs:NCName](#) [308]
Use: optional
Defined: locally [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ nillable

Type: [xs:boolean](#) [270]

Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

■ [ref](#)

Type: [xs:QName](#) [322]
Use: optional
Defined: locally [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ [substitutionGroup](#)

Type: [xs:QName](#) [322]
Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

■ [type](#)

Type: [xs:QName](#) [322]
Use: optional
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [405]

■ *{any attribute from non-schema namespace}*

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [404]

↔ [xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:keyref](#) [87]

Type: [anonymous](#) complexType (extension of [xs:keybase](#)) [88], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: locally within (this) [xs:element](#) complexType; see [XML source](#) [404]

↔ [xs:unique](#) [145]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

complexType xs:explicitGroup

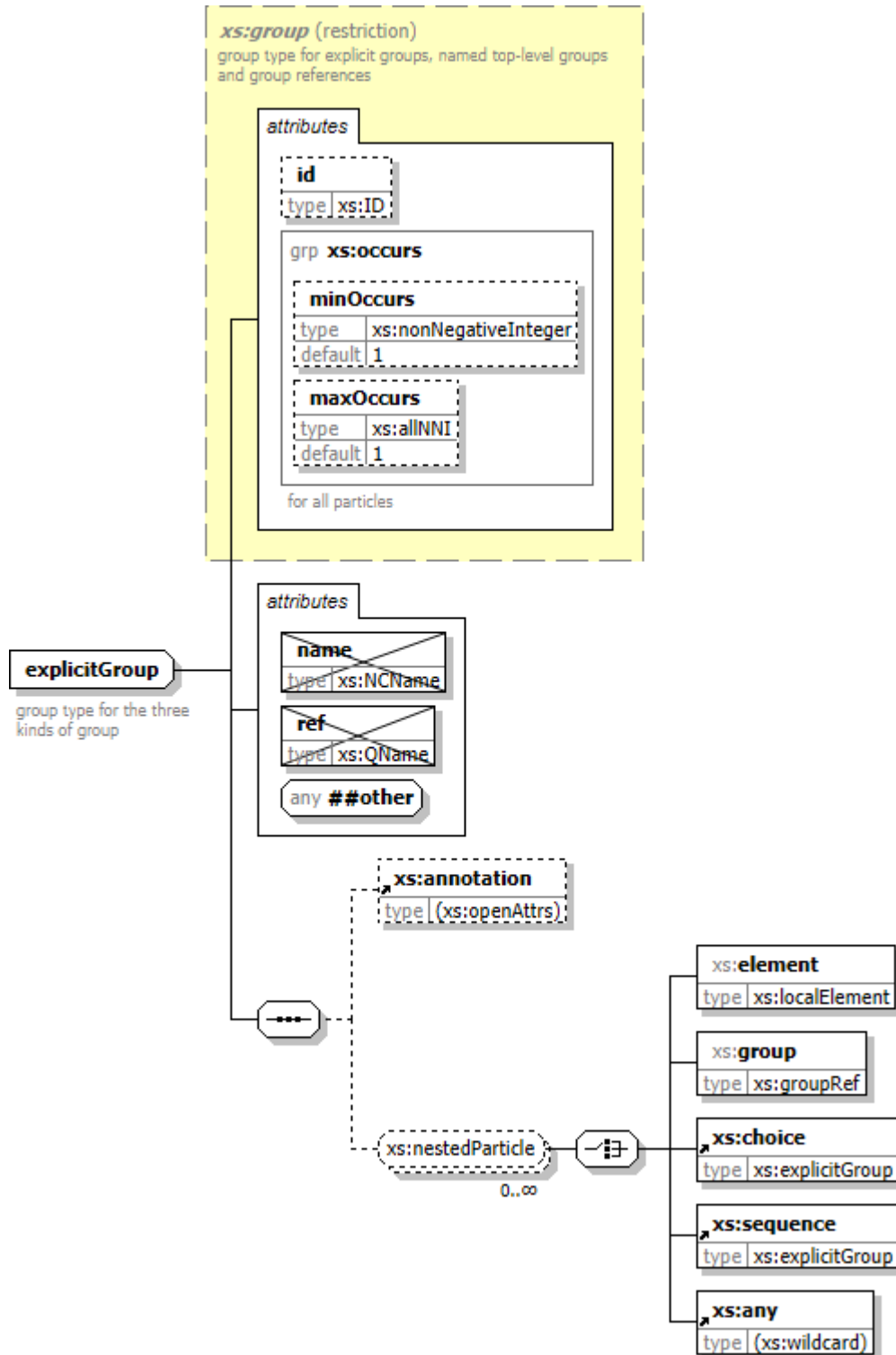
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 3 attributes, attr. wildcard, 6 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [180]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:element | xs:group | xs:choice | xs:sequence | xs:any)*
</...>
```

Content Model Elements (6):

[xs:annotation](#) [17], [xs:element](#) (type [xs:localElement](#)) [57],
[xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]

Known Direct Subtypes (2):

[xs:all](#) [150], [xs:simpleExplicitGroup](#) [233]

All Direct / Indirect Based Elements (6):

[xs:all](#) [12], [xs:choice](#) (in [xs:group](#)) [39],
[xs:all](#) (in [xs:group](#)) [15], [xs:sequence](#) [128],
[xs:choice](#) [36], [xs:sequence](#) (in [xs:group](#)) [131]

Known Usage Locations

- In derivations of other global types (2):

[xs:all](#)[150] (as restriction base), [xs:simpleExplicitGroup](#)[233] (as restriction base)

- As direct type of elements (2):

[xs:choice](#) [36], [xs:sequence](#) [128]

Annotation

group type for the three kinds of group

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   ├── xs:annotated [153] (extension)
│   │   └── xs:group [188] (restriction)
│   │       └── xs:explicitGroup
└──
```

XML Source (see within schema source: p. 406)

```
<xs:complexType name="explicitGroup">
  <xs:annotation>
    <xs:documentation>
      group type for the three kinds of group
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:restriction base="xs:group">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:nestedParticle"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```



```
<xs:attribute name="name" type="xs:NCName" use="prohibited"/>
<xs:attribute name="ref" type="xs:QName" use="prohibited"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 6/6)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Type: [xs:allNNI](#) [263]
Use: optional
Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

minOccurs

Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

name

Use: prohibited

ref

Use: prohibited

{any attribute from non-schema namespace}

Defined: [locally](#) within ([this](#)) [xs:explicitGroup](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 6/6)

xs:annotation [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:explicitGroup](#) complexType; see [XML source](#) [406]

xs:any [20]

Type: [anonymous](#) complexType ([extension of xs:wildcard](#)) [21], complex content
Defined: by reference [349] within [xs:nestedParticle](#) group; see [XML source](#) [400]

xs:choice [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: by reference [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content
Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content
Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

complexType xs:extensionType

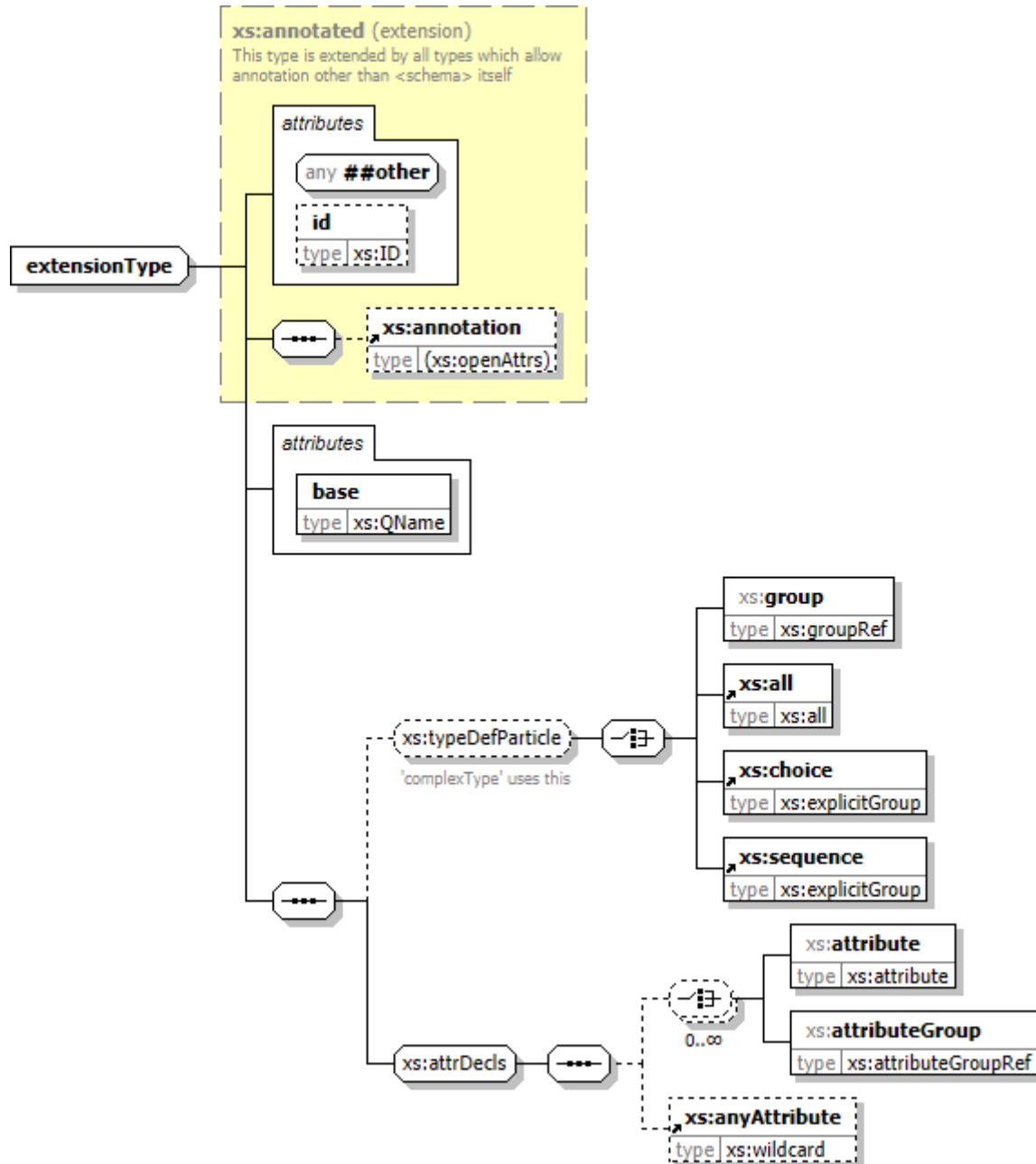
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 8 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [184]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:group | xs:all | xs:choice | xs:sequence)?, (xs:attribute |
  xs:attributeGroup)*, xs:anyAttribute?
</...>
```

Content Model Elements (8):

xs:all [12],	xs:attributeGroup (type xs:attributeGroupRef) [34],
xs:annotation [17],	xs:choice [36],
xs:anyAttribute [23],	xs:group (type xs:groupRef) [79],
xs:attribute (type xs:attribute) [29],	xs:sequence [128]

Known Direct Subtypes (1):

[xs:simpleExtensionType](#) [236]

All Direct / Indirect Based Elements (2):

[xs:extension](#) (in [xs:complexContent](#)) [67], [xs:extension](#) (in [xs:simpleContent](#)) [70]

Known Usage Locations

- In derivations of other global types (1):
 - [xs:simpleExtensionType](#) [236] (as restriction base)
- As direct type of elements (1):
 - [xs:extension](#) (in [xs:complexContent](#)) [67]

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├─ xs:openAttrs [224] (extension)
│   └─ xs:annotated [153] (extension)
│       └─ xs:extensionType
```

XML Source (see within schema source: p. 403)

```
<xs:complexType name="extensionType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:group minOccurs="0" ref="xs:typeDefParticle" />
        <xs:group ref="xs:attrDecls" />
      </xs:sequence>
      <xs:attribute name="base" type="xs:QName" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 3/3)

■ base

Type: [xs:QName](#) [322]
Use: required

Defined: locally within (this) [xs:extensionType](#) complexType; see [XML source](#) [403]

■ `id`

Type: [xs:ID](#) [295]

Use: optional

Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ *{any attribute from non-schema namespace}*

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 8/8)

↔ [xs:all](#) [12]

Type: [xs:all](#) [150], complex content

Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content

Defined: by reference [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content

Defined: by reference [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content

Defined: locally [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content

Defined: locally [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: locally [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

complexType xs:facet

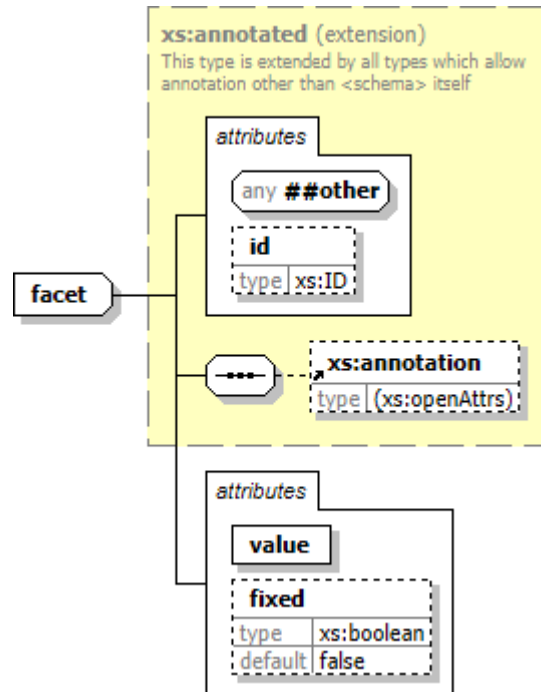
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 3 attributes, attr. wildcard, 1 element

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [187]

Component Diagram



XML Representation Summary

```
<...
  id      = xs:ID
  value   = xs:anySimpleType
  fixed   = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>
```

Content Model Elements (1):

[xs:annotation](#) [17]

Known Direct Subtypes (2):

[xs:noFixedFacet](#) [220], [xs:numFacet](#) [222]

All Direct / Indirect Based Elements (12):

[xs:enumeration](#) [65], [xs:fractionDigits](#) [75], [xs:length](#) [90], [xs:maxExclusive](#) [94], [xs:maxInclusive](#) [96], [xs:maxLength](#) [98], [xs:minExclusive](#) [100], [xs:minInclusive](#) [102], [xs:minLength](#) [104], [xs:pattern](#) [109], [xs:totalDigits](#) [140], [xs:whiteSpace](#) [147]

Known Usage Locations

- In derivations of other global types (2):

[xs:noFixedFacet](#) [220] (as restriction base), [xs:numFacet](#) [222] (as restriction base)

- **As direct type of elements (4):**
[xs:maxExclusive](#) [94], [xs:maxInclusive](#) [96], [xs:minExclusive](#) [100], [xs:minInclusive](#) [102]
- **In derivations of anonymous types of elements (1):**
[xs:whiteSpace](#) [147] (as restriction base)

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:facet
    
```

XML Source (see within schema source: p. 425)

```

<xs:complexType name="facet">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:attribute name="value" use="required"/>
      <xs:attribute default="false" name="fixed" type="xs:boolean" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: locally within (this) [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:anySimpleType](#)
Use: required
Defined: locally within (this) [xs:facet](#) complexType; see [XML source](#) [425]

{any attribute from non-schema namespace}

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

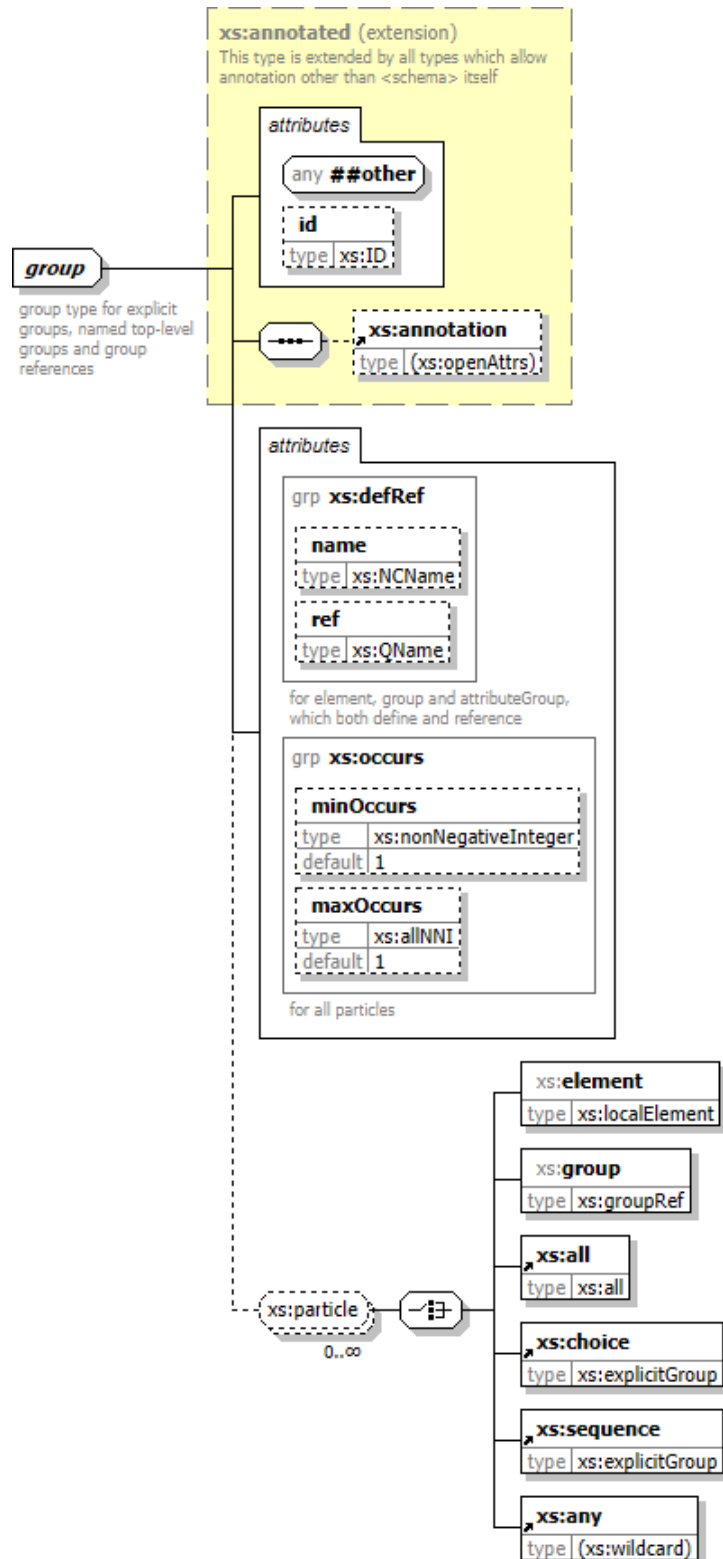
↔ [xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

complexType xs:group

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 5 attributes, attr. wildcard, 7 elements
Abstract: (cannot be assigned directly to elements used in instance XML documents)
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [189]

Component Diagram



XML Representation Summary

```
<...
  id           = xs:ID
  name        = xs:NCName
  ref         = xs:QName
  minOccurs   = xs:nonNegativeInteger : "1"
  maxOccurs   = (xs:nonNegativeInteger | "unbounded") : "1"
  {any attribute from non-schema namespace}
  >
  Content: xs:annotation?, (xs:element | xs:group | xs:all | xs:choice | xs:sequence | xs:any)*
</...>
```

Content Model Elements (7):

[xs:all](#) [12], [xs:element](#) (type [xs:localElement](#)) [57],
[xs:annotation](#) [17], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:any](#) [20], [xs:sequence](#) [128]
[xs:choice](#) [36],

Known Direct Subtypes (2):

[xs:explicitGroup](#) [179], [xs:realGroup](#) [226]

Known Indirect Subtypes (4):

[xs:all](#) [150], [xs:groupRef](#) [192], [xs:namedGroup](#) [212], [xs:simpleExplicitGroup](#) [233]

All Direct / Indirect Based Elements (8):

[xs:all](#) [12], [xs:group](#) [77],
[xs:all](#) (in [xs:group](#)) [15], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128],
[xs:choice](#) (in [xs:group](#)) [39], [xs:sequence](#) (in [xs:group](#)) [131]

Known Usage Locations

- In derivations of other global types (2):

[xs:explicitGroup](#) [179] (as restriction base), [xs:realGroup](#) [226] (as restriction base)

Annotation

group type for explicit groups, named top-level groups and group references

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:group
```

XML Source (see within schema source: p. 405)

```
<xs:complexType abstract="true" name="group">
  <xs:annotation>
    <xs:documentation>
      group type for explicit groups, named top-level groups and
      group references
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
```

```
<xs:extension base="xs:annotated">
  <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:particle"/>
  <xs:attributeGroup ref="xs:defRef"/>
  <xs:attributeGroup ref="xs:occurs"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 6/6)

id
 Type: [xs:ID](#) [295]
 Use: optional
 Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs
 Type: [xs:allNNI](#) [263]
 Use: optional
 Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

minOccurs
 Type: [xs:nonNegativeInteger](#) [314]
 Use: optional
 Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

name
 Type: [xs:NCName](#) [308]
 Use: optional
 Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

ref
 Type: [xs:QName](#) [322]
 Use: optional
 Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

{any attribute from non-schema namespace}
 Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 7/7)

xs:all [12]
 Type: [xs:all](#) [150], complex content
 Defined: [by reference](#) [351] within [xs:particle](#) group; see [XML source](#) [401]

xs:annotation [17]
 Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
 Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ `xs:any` [20]

Type: `anonymous complexType` (extension of `xs:wildcard`) [21], complex content
Defined: [by reference](#) [352] within `xs:particle` group; see [XML source](#) [401]

↔ `xs:choice` [36]

Type: `xs:explicitGroup` [179], complex content
Defined: [by reference](#) [352] within `xs:particle` group; see [XML source](#) [401]

↔ `xs:element` [57]

Type: `xs:localElement` [201], complex content
Defined: [locally](#) [352] within `xs:particle` group; see [XML source](#) [400]

↔ `xs:group` [79]

Type: `xs:groupRef` [192], complex content
Defined: [locally](#) [352] within `xs:particle` group; see [XML source](#) [400]

↔ `xs:sequence` [128]

Type: `xs:explicitGroup` [179], complex content
Defined: [by reference](#) [352] within `xs:particle` group; see [XML source](#) [401]

complexType xs:groupRef

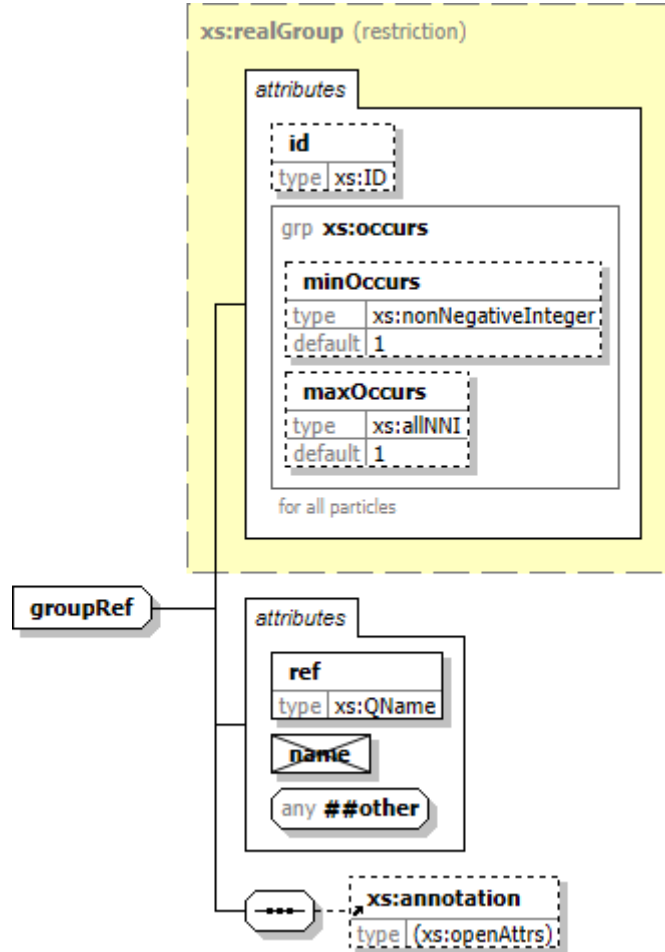
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 4 attributes, attr. wildcard, 1 element

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [193]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
  ref = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>
```

Content Model Elements (1):

[xs:annotation](#) [17]

All Direct / Indirect Based Elements (1):

[xs:group](#) (type [xs:groupRef](#)) [79]

Known Usage Locations

- As direct type of elements (1):

[xs:group](#) (type [xs:groupRef](#)) [79]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:group [188] (restriction)
│           └── xs:realGroup [226] (restriction)
│               └── xs:groupRef
    
```

XML Source (see within schema source: p. 406)

```

<xs:complexType name="groupRef">
  <xs:complexContent>
    <xs:restriction base="xs:realGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="ref" type="xs:QName" use="required"/>
      <xs:attribute name="name" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations: 6/6)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Type: [xs:allNNI](#) [263]
Use: optional
Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```
xs:nonNegativeInteger | "unbounded"
```

Default: "1"

minOccurs

Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

name

Use: prohibited

■ ref

Type: [xs:QName](#) [322]

Use: required

Defined: locally within ([this](#)) [xs:groupRef](#) complexType; see [XML source](#) [406]

■ *{any attribute from non-schema namespace}*

Defined: locally within ([this](#)) [xs:groupRef](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 1/1)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content

Defined: by reference within ([this](#)) [xs:groupRef](#) complexType; see [XML source](#) [406]

complexType xs:keybase

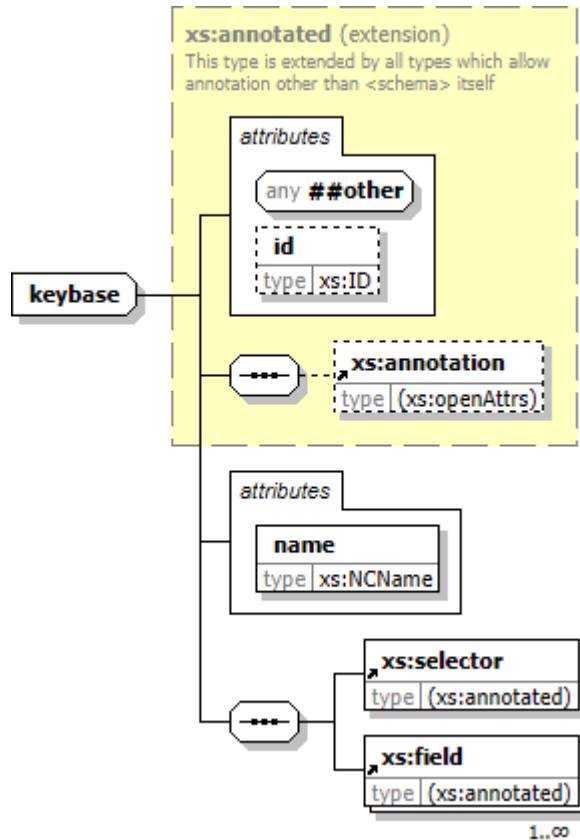
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 3 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [196]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:selector, xs:field+
</...>
```

Content Model Elements (3):

[xs:annotation](#) [17], [xs:field](#) [72], [xs:selector](#) [125]

All Direct / Indirect Based Elements (3):

[xs:key](#) [85], [xs:keyref](#) [87], [xs:unique](#) [145]

Known Usage Locations

- As direct type of elements (2):

[xs:key](#) [85], [xs:unique](#) [145]

- In derivations of anonymous types of elements (1):

[xs:keyref](#) [87] (as extension base)

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:keybase
    
```

XML Source (see within schema source: p. 411)

```

<xs:complexType name="keybase">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:element ref="xs:selector"/>
        <xs:element maxOccurs="unbounded" minOccurs="1" ref="xs:field"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 3/3)

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ name

Type: [xs:NCName](#) [308]
Use: required
Defined: [locally](#) within ([this](#)) [xs:keybase](#) complexType; see [XML source](#) [411]

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 3/3)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:field](#) [72]

Type: [anonymous](#) complexType (extension of [xs:annotated](#)) [73], complex content
Defined: [by reference](#) within ([this](#)) [xs:keybase](#) complexType; see [XML source](#) [411]

↔ [xs:selector](#) [125]

Type: [anonymous](#) complexType (extension of [xs:annotated](#)) [126], complex content
Defined: [by reference](#) within ([this](#)) [xs:keybase](#) complexType; see [XML source](#) [411]

complexType

xs:localComplexType

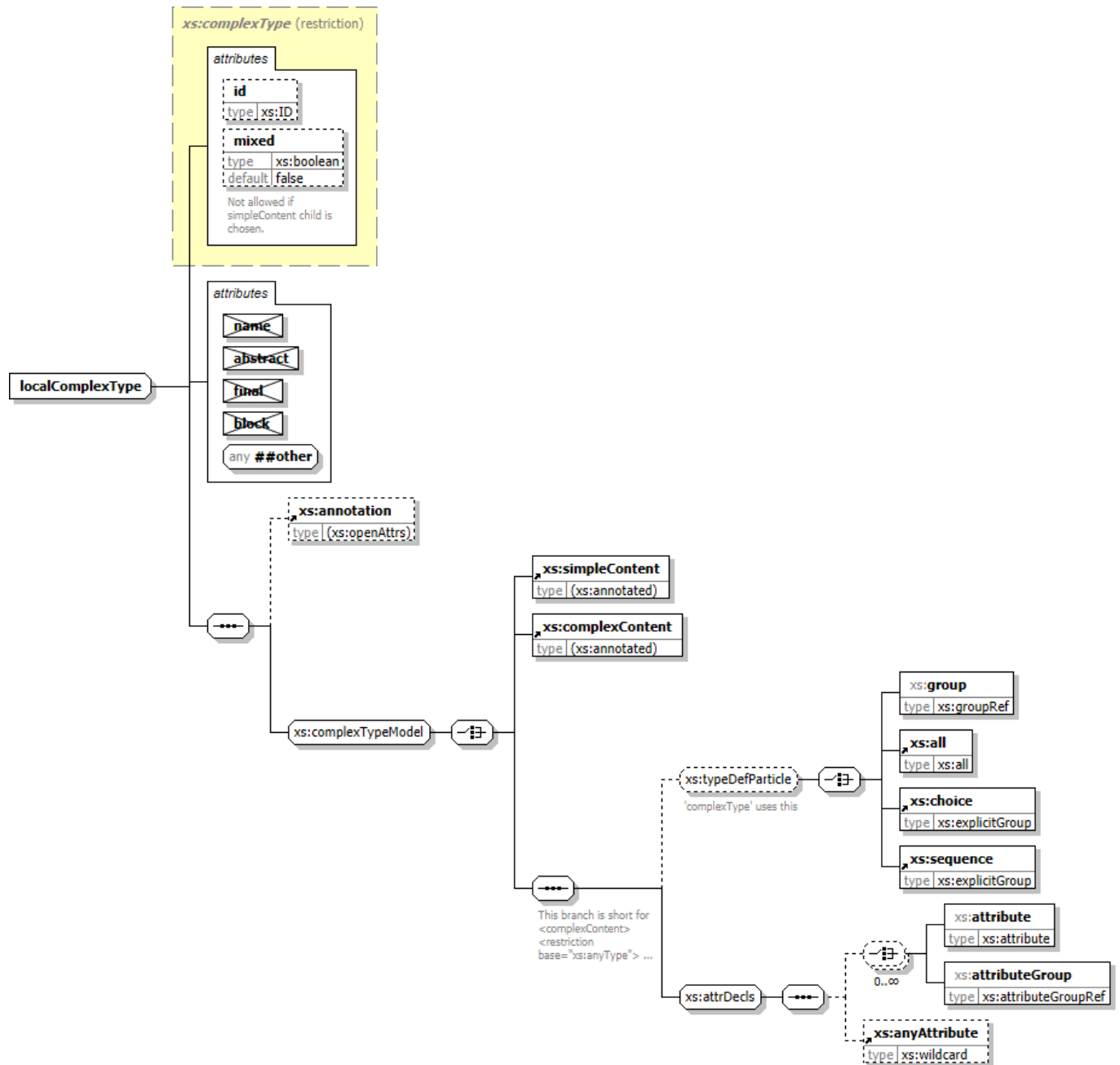
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 10 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [198]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  mixed = xs:boolean : "false"
  {any attribute from non-schema namespace}
>
  Content: xs:annotation?, (xs:simpleContent | xs:complexContent | ((xs:group | xs:all | xs:choice |
    xs:sequence)?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?))
</...>
```

Content Model Elements (10):

- | | |
|--|--|
| xs:all [12], | xs:choice [36], |
| xs:annotation [17], | xs:complexContent [41], |
| xs:anyAttribute [23], | xs:group (type xs:groupRef) [79], |
| xs:attribute (type xs:attribute) [29], | xs:sequence [128], |
| xs:attributeGroup (type xs:attributeGroupRef) [34], | xs:simpleContent [133] |

All Direct / Indirect Based Elements (1):

- xs:complexType (type [xs:localComplexType](#)) [48]

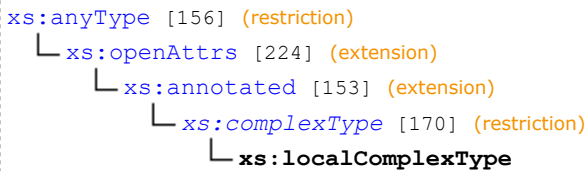
Known Usage Locations

- As direct type of elements (1):

- xs:complexType (type [xs:localComplexType](#)) [48]

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 402)

```
<xs:complexType name="localComplexType">
  <xs:complexContent>
    <xs:restriction base="xs:complexType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:complexTypeModel"/>
      </xs:sequence>
      <xs:attribute name="name" use="prohibited"/>
      <xs:attribute name="abstract" use="prohibited"/>
      <xs:attribute name="final" use="prohibited"/>
      <xs:attribute name="block" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 7/7)

abstract

Use: prohibited

■ ~~block~~

Use: prohibited

■ ~~final~~

Use: prohibited

■ id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ mixed

Type: [xs:boolean](#) [270]

Use: optional

Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Not allowed if simpleContent child is chosen.

May be overridden by setting on complexContent child.

Attribute Value

Default: "false"

■ ~~name~~

Use: prohibited

■ {any attribute from non-schema namespace}

Defined: [locally](#) within ([this](#)) [xs:localComplexType](#) complexType; see [XML source](#) [402]

Content Element Detail (all declarations; 10/10)

↔ [xs:all](#) [12]

Type: [xs:all](#) [150], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content

Defined: [by reference](#) within ([this](#)) [xs:localComplexType](#) complexType; see [XML source](#) [402]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content

Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content

Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content

Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ `xs:complexContent` [41]

Type: `anonymous` complexType (extension of `xs:annotated`) [42], complex content
Defined: `by reference` [343] within `xs:complexTypeModel` group; see [XML source](#) [401]

↔ `xs:group` [79]

Type: `xs:groupRef` [192], complex content
Defined: `locally` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:sequence` [128]

Type: `xs:explicitGroup` [179], complex content
Defined: `by reference` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

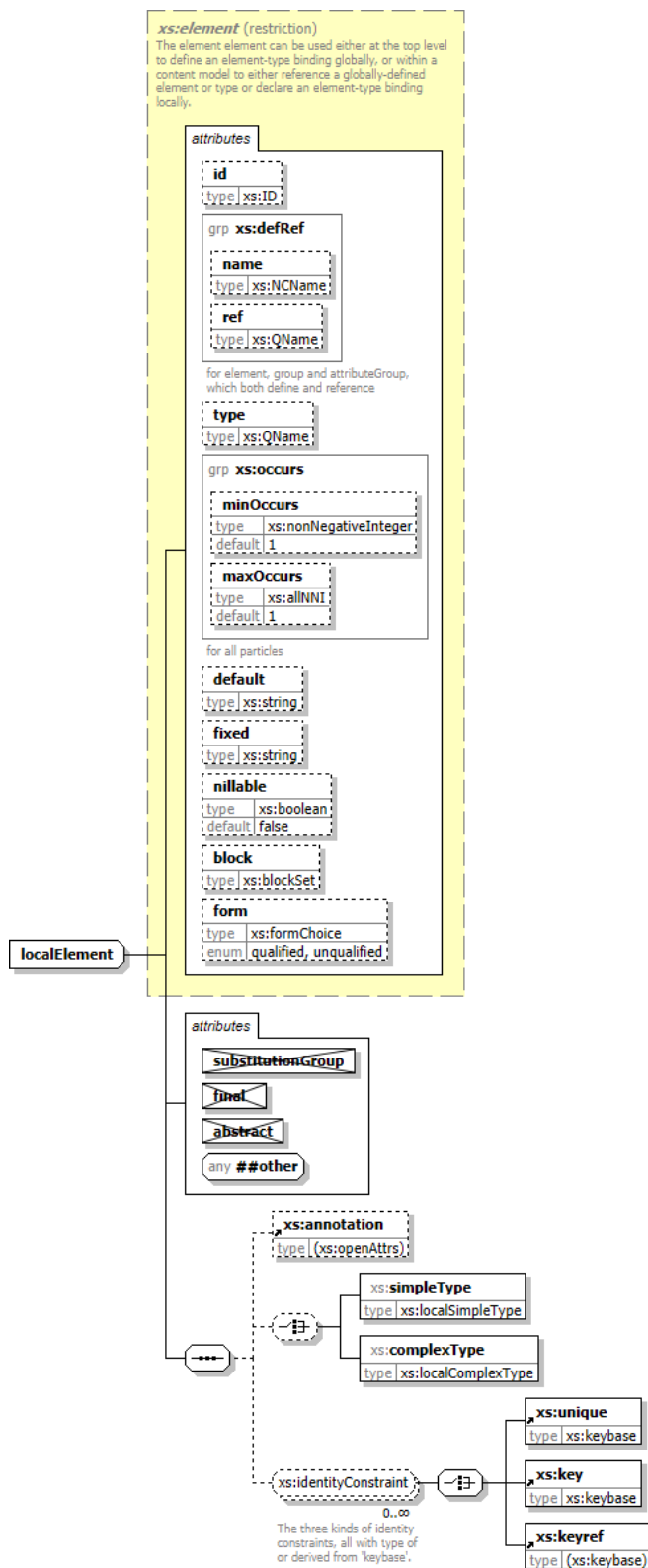
↔ `xs:simpleContent` [133]

Type: `anonymous` complexType (extension of `xs:annotated`) [134], complex content
Defined: `by reference` [343] within `xs:complexTypeModel` group; see [XML source](#) [401]

complexType xs:localElement

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 11 attributes, attr. wildcard, 6 elements
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [202]

Component Diagram



XML Representation Summary

```

<...
  id           = xs:ID
  name        = xs:NCName
  ref         = xs:QName
  type        = xs:QName
  minOccurs   = xs:nonNegativeInteger : "1"
  maxOccurs   = (xs:nonNegativeInteger | "unbounded") : "1"
  default     = xs:string
  fixed       = xs:string
  nillable    = xs:boolean : "false"
  block       = ("#all" | list of ("extension" | "restriction" | "substitution"))
  form        = ("qualified" | "unqualified")
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</...>

```

Content Model Elements (6):

[xs:annotation](#) [17], [xs:keyref](#) [87],
[xs:complexType](#) (type [xs:localComplexType](#)) [48], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
[xs:key](#) [85], [xs:unique](#) [145]

Known Direct Subtypes (1):

[xs:narrowMaxMin](#) [215]

All Direct / Indirect Based Elements (2):

[xs:element](#) (type [xs:localElement](#)) [57], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

Known Usage Locations

- In derivations of other global types (1):
 - [xs:narrowMaxMin](#) [215] (as restriction base)
- As direct type of elements (1):
 - [xs:element](#) (type [xs:localElement](#)) [57]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:element [174] (restriction)
│           └── xs:localElement

```

XML Source (see within schema source: p. 405)

```

<xs:complexType name="localElement">
  <xs:complexContent>
    <xs:restriction base="xs:element">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:attribute name="substitutionGroup" use="prohibited"/>
<xs:attribute name="final" use="prohibited"/>
<xs:attribute name="abstract" use="prohibited"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>

```

Attribute Detail (all declarations; 15/15)

abstract

Use: prohibited

block

Type: [xs:blockSet](#) [268]
 Use: optional
 Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```

"#all" | list of ("extension" | "restriction" | "substitution")

```

default

Type: [xs:string](#) [328]
 Use: optional
 Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

final

Use: prohibited

fixed

Type: [xs:string](#) [328]
 Use: optional
 Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

form

Type: [xs:formChoice](#) [286]
 Use: optional
 Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```

enumeration of xs:NMTOKEN

```

Enumeration: "qualified", "unqualified"

id

Type: [xs:ID](#) [295]
 Use: optional
 Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Type: [xs:allNNI](#) [263]
 Use: optional
 Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

```

xs:nonNegativeInteger | "unbounded"

```

Default: "1"

■ minOccurs

Type: [xs:nonNegativeInteger](#) [314]
Use: optional
Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

■ name

Type: [xs:NCName](#) [308]
Use: optional
Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ nillable

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

■ ref

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ substitutionGroup

Use: prohibited

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ {any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:localElement](#) complexType; see [XML source](#) [405]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:localElement](#) complexType; see [XML source](#) [405]

↔ [xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content
Defined: locally within ([this](#)) [xs:localElement](#) complexType; see [XML source](#) [405]

↔ [xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:keyref](#) [87]

Type: [anonymous](#) complexType ([extension of](#) [xs:keybase](#)) [88], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content

Defined: locally within ([this](#)) [xs:localElement](#) complexType; see [XML source](#) [405]

↔ [xs:unique](#) [145]

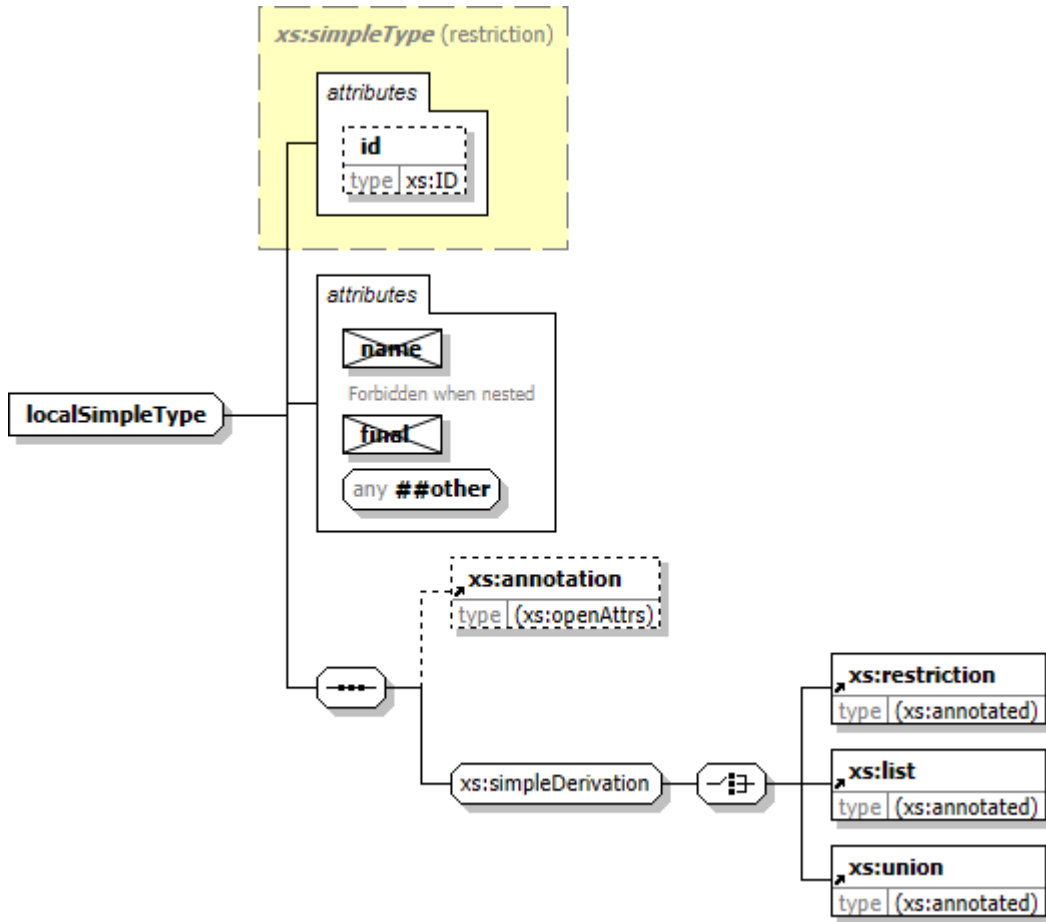
Type: [xs:keybase](#) [195], complex content

Defined: [by reference](#) [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

complexType xs:localSimpleType

Namespace: <http://www.w3.org/2001/XMLSchema>
 Content: complex, 1 attribute, attr. wildcard, 4 elements
 Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [207]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:restriction | xs:list | xs:union)
</...>
```

Content Model Elements (4):

[xs:annotation](#) [17], [xs:list](#) [92], [xs:restriction](#) [114], [xs:union](#) [142]

All Direct / Indirect Based Elements (1):

[xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Known Usage Locations

- As direct type of elements (1):

[xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:simpleType [243] (restriction)
│           └── xs:localSimpleType
    
```

XML Source (see within schema source: p. 424)

```

<xs:complexType name="localSimpleType">
  <xs:complexContent>
    <xs:restriction base="xs:simpleType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:simpleDerivation"/>
      </xs:sequence>
      <xs:attribute name="name" use="prohibited">
        <xs:annotation>
          <xs:documentation>
            Forbidden when nested
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="final" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

- `final`
 - Use: prohibited
- `id`
 - Type: `xs:ID` [295]
 - Use: optional
 - Defined: locally [155] within `xs:annotated` complexType; see [XML source](#) [397]
- `name`
 - Use: prohibited
 - Forbidden when nested
- `{any attribute from non-schema namespace}`
 - Defined: locally within (this) `xs:localSimpleType` complexType; see [XML source](#) [424]

Content Element Detail (all declarations; 4/4)

- ↔ `xs:annotation` [17]
 - Type: anonymous complexType (extension of `xs:openAttrs`) [18], complex content
 - Defined: by reference within (this) `xs:localSimpleType` complexType; see [XML source](#) [424]
- ↔ `xs:list` [92]
 - Type: anonymous complexType (extension of `xs:annotated`) [93], complex content
 - Defined: by reference [357] within `xs:simpleDerivation` group; see [XML source](#) [423]

↔ [xs:restriction](#) [114]

Type: [anonymous complexType](#) ([extension of xs:annotated](#)) [115], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

↔ [xs:union](#) [142]

Type: [anonymous complexType](#) ([extension of xs:annotated](#)) [143], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

complexType xs:namedAttributeGroup

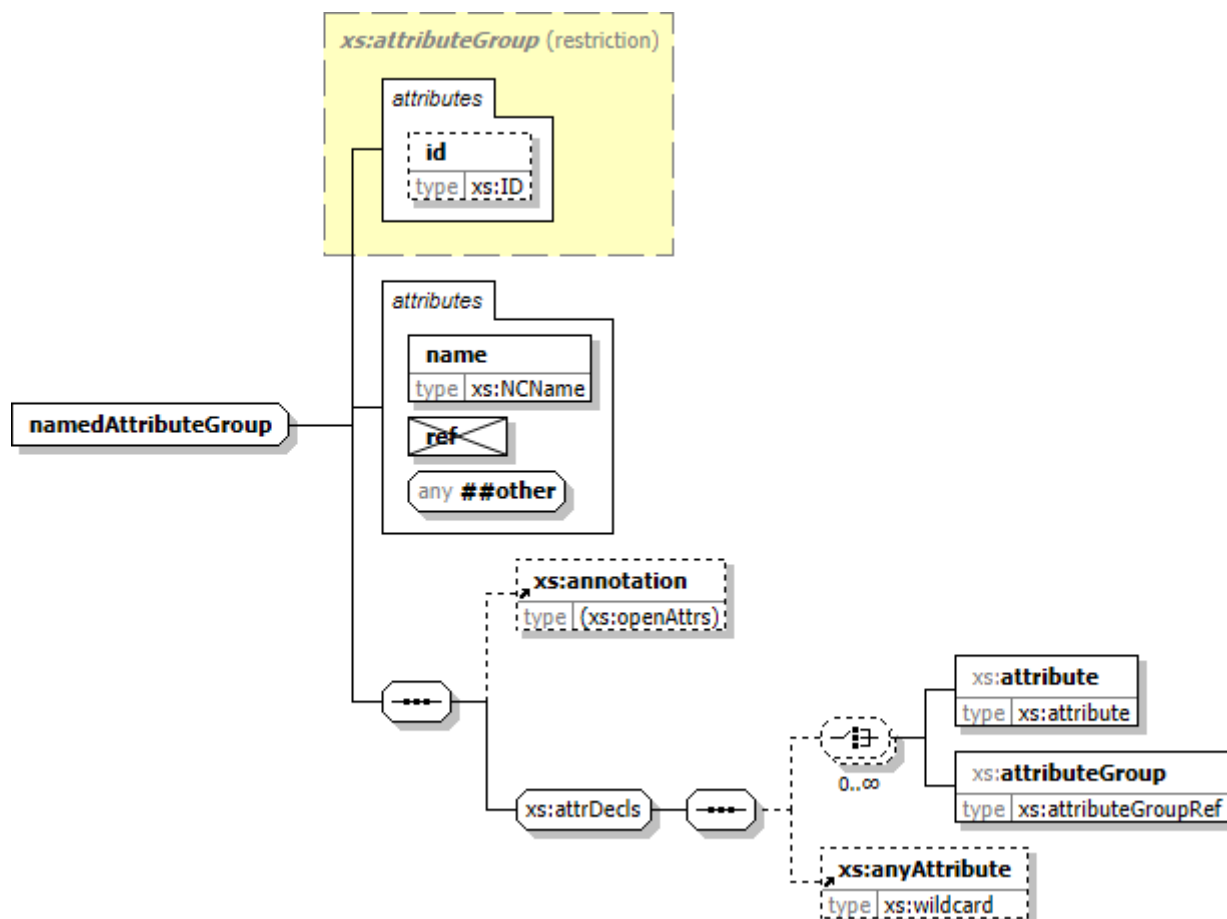
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 4 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [210]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?
</...>
```

Content Model Elements (4):

[xs:annotation](#) [17], [xs:attribute](#) (type [xs:attribute](#)) [29],
[xs:anyAttribute](#) [23], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

All Direct / Indirect Based Elements (1):

[xs:attributeGroup](#) [32]

Known Usage Locations

- As direct type of elements (1):

[xs:attributeGroup](#) [32]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:attributeGroup [162] (restriction)
│           └── xs:namedAttributeGroup
    
```

XML Source (see within schema source: p. 409)

```

<xs:complexType name="namedAttributeGroup">
  <xs:complexContent>
    <xs:restriction base="xs:attributeGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:attrDecls"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: locally within (this) [xs:namedAttributeGroup](#) complexType; see [XML source](#) [409]

ref

Use: prohibited

{any attribute from non-schema namespace}

Defined: locally within (this) [xs:namedAttributeGroup](#) complexType; see [XML source](#) [409]

Content Element Detail (all declarations; 4/4)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference within (this) [xs:namedAttributeGroup](#) complexType; see [XML source](#) [409]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content

Defined: [by reference](#) [340] within `xs:attrDecls` group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: `xs:attribute` [158], complex content

Defined: [locally](#) [340] within `xs:attrDecls` group; see [XML source](#) [401]

↔ [xs:attributeGroup](#) [34]

Type: `xs:attributeGroupRef` [165], complex content

Defined: [locally](#) [341] within `xs:attrDecls` group; see [XML source](#) [401]

complexType xs:namedGroup

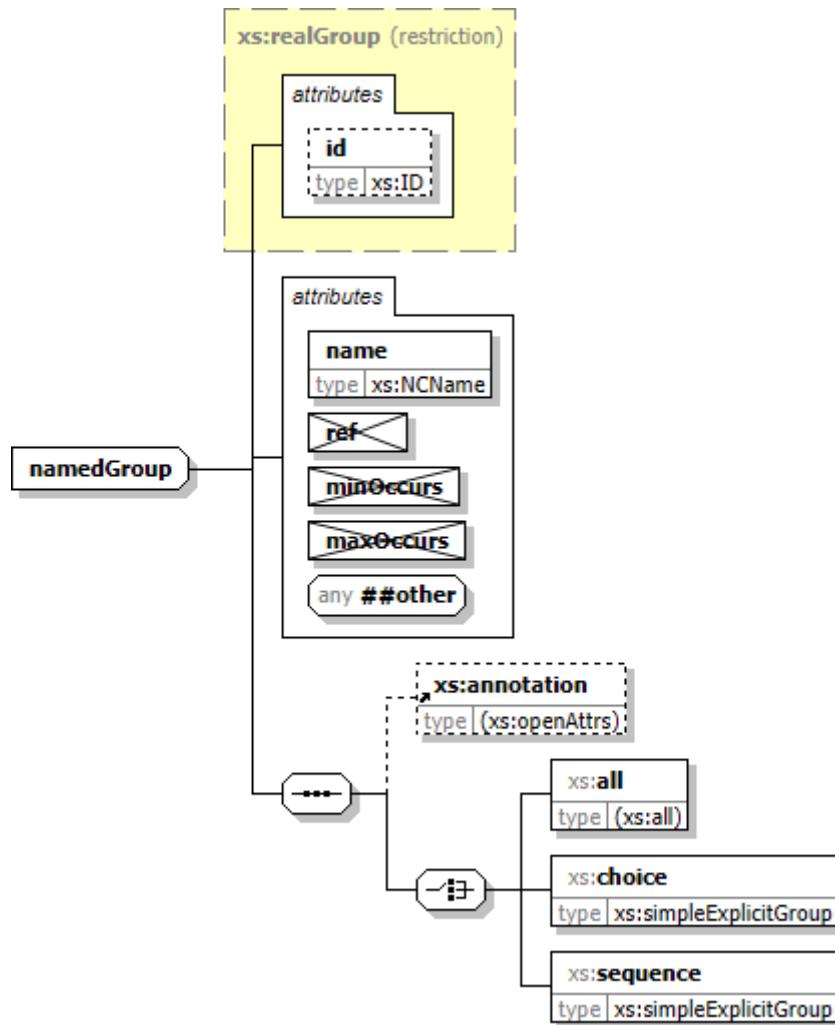
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 4 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [213]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:all | xs:choice | xs:sequence)
</...>

```

Content Model Elements (4):

[xs:all](#) (in [xs:group](#)) [15], [xs:choice](#) (in [xs:group](#)) [39],
[xs:annotation](#) [17], [xs:sequence](#) (in [xs:group](#)) [131]

All Direct / Indirect Based Elements (1):

[xs:group](#) [77]

Known Usage Locations

- As direct type of elements (1):

[xs:group](#) [77]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:group [188] (restriction)
│           └── xs:realGroup [226] (restriction)
│               └── xs:namedGroup
    
```

XML Source (see within schema source: p. 406)

```

<xs:complexType name="namedGroup">
  <xs:complexContent>
    <xs:restriction base="xs:realGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice maxOccurs="1" minOccurs="1">
          <xs:element name="all">
            <xs:complexType>
              <xs:complexContent>
                <xs:restriction base="xs:all">
                  <xs:group ref="xs:allModel"/>
                  <xs:attribute name="minOccurs" use="prohibited"/>
                  <xs:attribute name="maxOccurs" use="prohibited"/>
                  <xs:anyAttribute namespace="##other" processContents="lax"/>
                </xs:restriction>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="choice" type="xs:simpleExplicitGroup"/>
          <xs:element name="sequence" type="xs:simpleExplicitGroup"/>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:attribute name="minOccurs" use="prohibited"/>
      <xs:attribute name="maxOccurs" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations: 6/6)

■ `id`

Type: `xs:ID` [295]
Use: optional
Defined: `locally` [155] within `xs:annotated` complexType; see [XML source](#) [397]

■ `maxOccurs`

Use: prohibited

■ `minOccurs`

Use: prohibited

■ name

Type: [xs:NCName](#) [308]

Use: required

Defined: locally within ([this](#)) [xs:namedGroup](#) complexType; see [XML source](#) [406]

■ ~~ref~~

Use: prohibited

■ {any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:namedGroup](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 4/4)

↔ [xs:all](#) [15]

Type: [anonymous](#) complexType ([restriction of xs:all](#)) [16], complex content

Defined: locally within ([this](#)) [xs:namedGroup](#) complexType; see [XML source](#) [406]

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content

Defined: by reference within ([this](#)) [xs:namedGroup](#) complexType; see [XML source](#) [406]

↔ [xs:choice](#) [39]

Type: [xs:simpleExplicitGroup](#) [233], complex content

Defined: locally within ([this](#)) [xs:namedGroup](#) complexType; see [XML source](#) [406]

↔ [xs:sequence](#) [131]

Type: [xs:simpleExplicitGroup](#) [233], complex content

Defined: locally within ([this](#)) [xs:namedGroup](#) complexType; see [XML source](#) [406]

complexType xs:narrowMaxMin

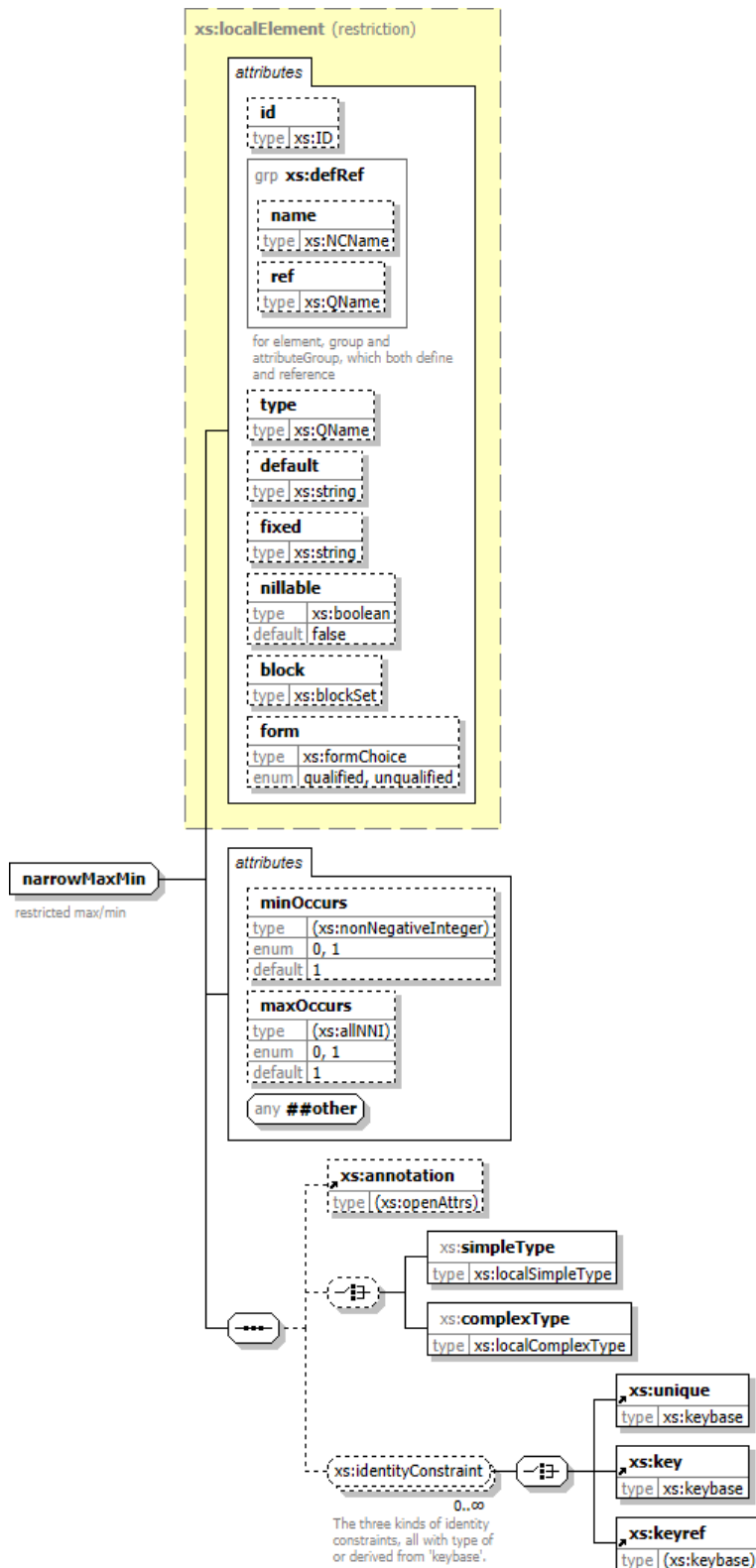
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 11 attributes, attr. wildcard, 6 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [216]

Component Diagram



XML Representation Summary

```
<...
  id           = xs:ID
  name        = xs:NCName
  ref         = xs:QName
  type        = xs:QName
  default     = xs:string
  fixed       = xs:string
  nillable    = xs:boolean : "false"
  block      = ("#all" | list of ("extension" | "restriction" | "substitution"))
  form       = ("qualified" | "unqualified")
  minOccurs   = ("0" | "1") : "1"
  maxOccurs   = ("0" | "1") : "1"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</...>
```

Content Model Elements (6):

- [xs:annotation](#) [17],
- [xs:complexType](#) (type [xs:localComplexType](#)) [48],
- [xs:key](#) [85],
- [xs:keyref](#) [87],
- [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
- [xs:unique](#) [145]

All Direct / Indirect Based Elements (1):

- [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

Known Usage Locations

- As direct type of elements (1):

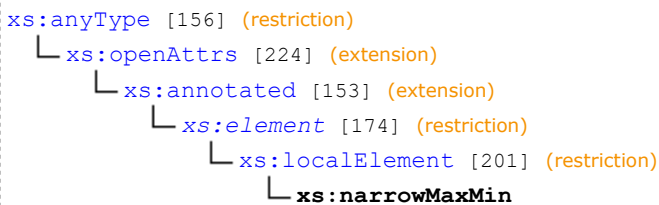
- [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

Annotation

restricted max/min

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 407)

```
<xs:complexType name="narrowMaxMin">
  <xs:annotation>
    <xs:documentation>restricted max/min</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:restriction base="xs:localElement">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

```

</xs:sequence>
<xs:attribute default="1" name="minOccurs" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:nonNegativeInteger">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute default="1" name="maxOccurs" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:allNNI">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>

```

Attribute Detail (all declarations; 12/12)

block

Type: [xs:blockSet](#) [268]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "substitution")
```

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

form

Type: [xs:formChoice](#) [286]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
enumeration of xs:NMTOKEN
```

Enumeration: "qualified", "unqualified"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Type: [anonymous](#) simpleType (restriction of [xs:allNNI](#)) [218]
Use: optional
Defined: [locally](#) within ([this](#)) [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

Attribute Value

```
enumeration of (xs:nonNegativeInteger | "unbounded")
```

Enumeration: "0", "1"

Default: "1"

Anonymous simpleType

Type Derivation Tree

```
union of (xs:nonNegativeInteger | restriction of xs:NMTOKEN)
```

```
└─ xs:allNNI [263] (restriction)
```

```
└─ simpleType
```

minOccurs

Type: anonymous simpleType (restriction of xs:nonNegativeInteger) [218]

Use: optional

Defined: locally within (this) xs:narrowMaxMin complexType; see XML source [407]

Attribute Value

```
enumeration of xs:nonNegativeInteger
```

Enumeration: "0", "1"

Default: "1"

Anonymous simpleType

Type Derivation Tree

```
xs:anySimpleType (restriction)
```

```
└─ xs:decimal [275] (restriction)
```

```
└─ xs:integer [299] (restriction)
```

```
└─ xs:nonNegativeInteger [314] (restriction)
```

```
└─ simpleType
```

name

Type: xs:NCName [308]

Use: optional

Defined: locally [363] within xs:defRef attributeGroup; see XML source [400]

nillable

Type: xs:boolean [270]

Use: optional

Defined: locally [177] within xs:element complexType; see XML source [405]

Attribute Value

Default: "false"

ref

Type: xs:QName [322]

Use: optional

Defined: locally [364] within xs:defRef attributeGroup; see XML source [400]

type

Type: xs:QName [322]

Use: optional

Defined: locally [178] within xs:element complexType; see XML source [405]

{any attribute from non-schema namespace}

Defined: locally within (this) xs:narrowMaxMin complexType; see XML source [407]

Content Element Detail (all declarations; 6/6)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

[xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content
Defined: locally within ([this](#)) [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

[xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

[xs:keyref](#) [87]

Type: [anonymous](#) complexType ([extension of xs:keybase](#)) [88], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

[xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: locally within ([this](#)) [xs:narrowMaxMin](#) complexType; see [XML source](#) [407]

[xs:unique](#) [145]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

complexType xs:noFixedFacet

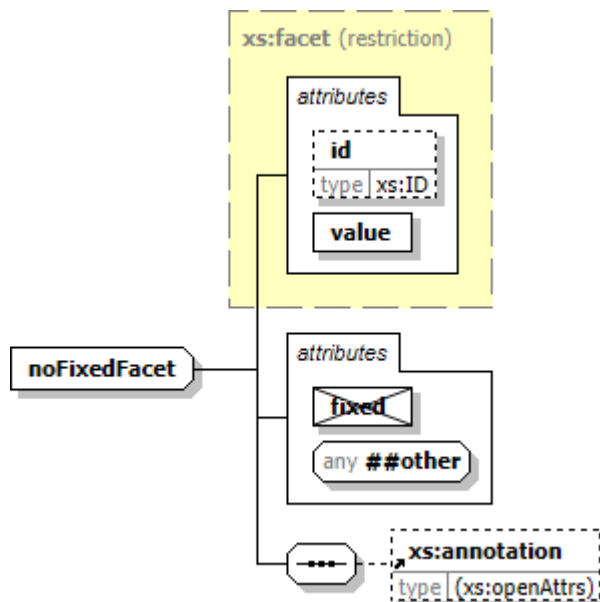
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 1 element

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [221]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  value = xs:anySimpleType
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>
```

Content Model Elements (1):

[xs:annotation](#) [17]

All Direct / Indirect Based Elements (2):

[xs:enumeration](#) [65], [xs:pattern](#) [109]

Known Usage Locations

- As direct type of elements (1):
[xs:enumeration](#) [65]
- In derivations of anonymous types of elements (1):
[xs:pattern](#) [109] (as restriction base)

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:facet [186] (restriction)
│           └── xs:noFixedFacet
    
```

XML Source (see within schema source: p. 425)

```

<xs:complexType name="noFixedFacet">
  <xs:complexContent>
    <xs:restriction base="xs:facet">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="fixed" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

fixed

Use: prohibited

id

Type: [xs:ID](#) [295]

Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:anySimpleType](#)

Use: required

Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

{any attribute from non-schema namespace}

Defined: [locally](#) within ([this](#)) [xs:noFixedFacet](#) complexType; see [XML source](#) [425]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content

Defined: by reference within ([this](#)) [xs:noFixedFacet](#) complexType; see [XML source](#) [425]

complexType xs:numFacet

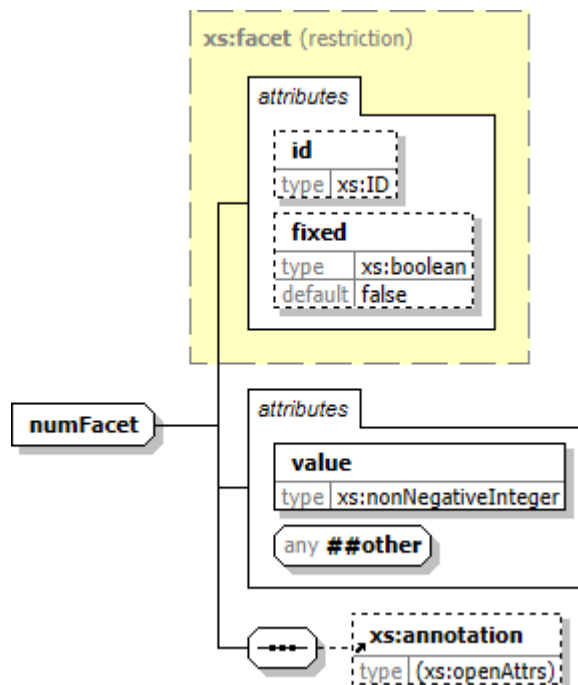
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 3 attributes, attr. wildcard, 1 element

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [223]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  fixed = xs:boolean : "false"
  value = xs:nonNegativeInteger
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>

```

Content Model Elements (1):

[xs:annotation](#) [17]

All Direct / Indirect Based Elements (5):

[xs:fractionDigits](#) [75], [xs:length](#) [90], [xs:maxLength](#) [98], [xs:minLength](#) [104], [xs:totalDigits](#) [140]

Known Usage Locations

- As direct type of elements (4):

[xs:fractionDigits](#) [75], [xs:length](#) [90], [xs:maxLength](#) [98], [xs:minLength](#) [104]

- In derivations of anonymous types of elements (1):

[xs:totalDigits](#) [140] (as restriction base)

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   ├── xs:annotated [153] (extension)
│   │   └── xs:facet [186] (restriction)
│   │       └── xs:numFacet

```

XML Source (see within schema source: p. 426)

```

<xs:complexType name="numFacet">
  <xs:complexContent>
    <xs:restriction base="xs:facet">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="value" type="xs:nonNegativeInteger" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

Attribute Detail (all declarations; 4/4)

fixed

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [187] within [xs:facet](#) complexType; see [XML source](#) [425]

Attribute Value

Default: "false"

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

value

Type: [xs:nonNegativeInteger](#) [314]
Use: required
Defined: locally within ([this](#)) [xs:numFacet](#) complexType; see [XML source](#) [426]

{any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:numFacet](#) complexType; see [XML source](#) [426]

Content Element Detail (all declarations; 1/1)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:numFacet](#) complexType; see [XML source](#) [426]

complexType

xs:openAttrs

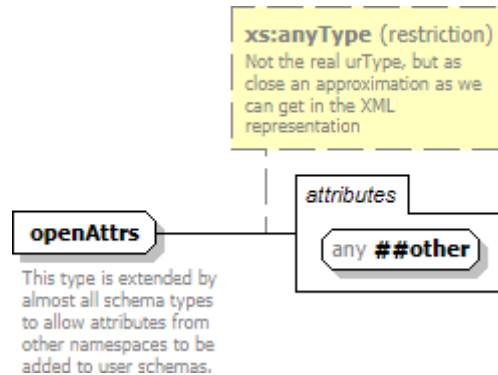
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: empty, attr. [wildcard](#)

Block: "#all" (*blocks all substitutions of this complex type through xsi:type attribute in instance XML documents*)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [225]

Component Diagram



XML Representation Summary

```
<...
  {any attribute from non-schema namespace}
/>
```

Known Direct Subtypes (1):

[xs:annotated](#) [153]

Known Indirect Subtypes (32):

[xs:all](#) [150], [xs:attribute](#) [158], [xs:attributeGroup](#) [162], [xs:attributeGroupRef](#) [165], [xs:complexRestrictionType](#) [167], [xs:complexType](#) [170], [xs:element](#) [174], [xs:explicitGroup](#) [179], [xs:extensionType](#) [183], [xs:facet](#) [186], [xs:group](#) [188], [xs:groupRef](#) [192], [xs:keybase](#) [195], [xs:localComplexType](#) [197], [xs:localElement](#) [201], [xs:localSimpleType](#) [206], [xs:namedAttributeGroup](#) [209], [xs:namedGroup](#) [212], [xs:narrowMaxMin](#) [215], [xs:noFixedFacet](#) [220], [xs:numFacet](#) [222], [xs:realGroup](#) [226], [xs:restrictionType](#) [229], [xs:simpleExplicitGroup](#) [233], [xs:simpleExtensionType](#) [236], [xs:simpleRestrictionType](#) [239], [xs:simpleType](#) [243], [xs:topLevelAttribute](#) [246], [xs:topLevelComplexType](#) [249], [xs:topLevelElement](#) [253], [xs:topLevelSimpleType](#) [257], [xs:wildcard](#) [260]

All Direct / Indirect Based Elements (53):

[xs:all](#) [12],
[xs:all](#) (in [xs:group](#)) [15],
[xs:annotation](#) [17],
[xs:any](#) [20],
[xs:anyAttribute](#) [23],
[xs:attribute](#) [27],
[xs:attribute](#) (type [xs:attribute](#)) [29],
[xs:attributeGroup](#) [32],
[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34],
[xs:choice](#) [36],
[xs:choice](#) (in [xs:group](#)) [39],
[xs:complexContent](#) [41],
[xs:complexType](#) [44],
[xs:complexType](#) (type [xs:localComplexType](#)) [48],
[xs:element](#) [53],
[xs:element](#) (type [xs:localElement](#)) [57],
[xs:element](#) (type [xs:narrowMaxMin](#)) [61],
[xs:keyref](#) [87],
[xs:length](#) [90],
[xs:list](#) [92],
[xs:maxExclusive](#) [94],
[xs:maxInclusive](#) [96],
[xs:maxLength](#) [98],
[xs:minExclusive](#) [100],
[xs:minInclusive](#) [102],
[xs:minLength](#) [104],
[xs:notation](#) [106],
[xs:pattern](#) [109],
[xs:redefine](#) [111],
[xs:restriction](#) [114],
[xs:restriction](#) (in [xs:complexContent](#)) [118],
[xs:restriction](#) (in [xs:simpleContent](#)) [121],
[xs:schema](#) [7],
[xs:selector](#) [125],

- [xs:enumeration](#) [65],
- [xs:extension](#) (in [xs:complexContent](#)) [67],
- [xs:extension](#) (in [xs:simpleContent](#)) [70],
- [xs:field](#) [72],
- [xs:fractionDigits](#) [75],
- [xs:group](#) [77],
- [xs:group](#) (type [xs:groupRef](#)) [79],
- [xs:import](#) [81],
- [xs:include](#) [83],
- [xs:key](#) [85],
- [xs:sequence](#) [128],
- [xs:sequence](#) (in [xs:group](#)) [131],
- [xs:simpleContent](#) [133],
- [xs:simpleType](#) [135],
- [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
- [xs:totalDigits](#) [140],
- [xs:union](#) [142],
- [xs:unique](#) [145],
- [xs:whiteSpace](#) [147]

Known Usage Locations

- In derivations of other global types (1):**
 - [xs:annotated](#) [153] (as extension base)
- In derivations of anonymous types of elements (3):**
 - [xs:annotation](#) [17] (as extension base), [xs:schema](#) [7] (as extension base)
 - [xs:redefine](#) [111] (as extension base),

Annotation

This type is extended by almost all schema types to allow attributes from other namespaces to be added to user schemas.

Type Definition Detail

Type Derivation Tree

[xs:anyType](#) [156] (restriction)
└─ [xs:openAttrs](#)

XML Source (see within schema source: p. 397)

```
<xs:complexType name="openAttrs">  
  <xs:annotation>  
    <xs:documentation>  
      This type is extended by almost all schema types  
to allow attributes from other namespaces to be  
added to user schemas.  
    </xs:documentation>  
  </xs:annotation>  
  <xs:complexContent>  
    <xs:restriction base="xs:anyType">  
      <xs:anyAttribute namespace="##other" processContents="lax"/>  
    </xs:restriction>  
  </xs:complexContent>  
</xs:complexType>
```

Attribute Detail (all declarations; 1/1)

■ {any attribute from non-schema namespace}

Defined: locally within (this) [xs:openAttrs](#) complexType; see [XML source](#) [397]

complexType xs:realGroup

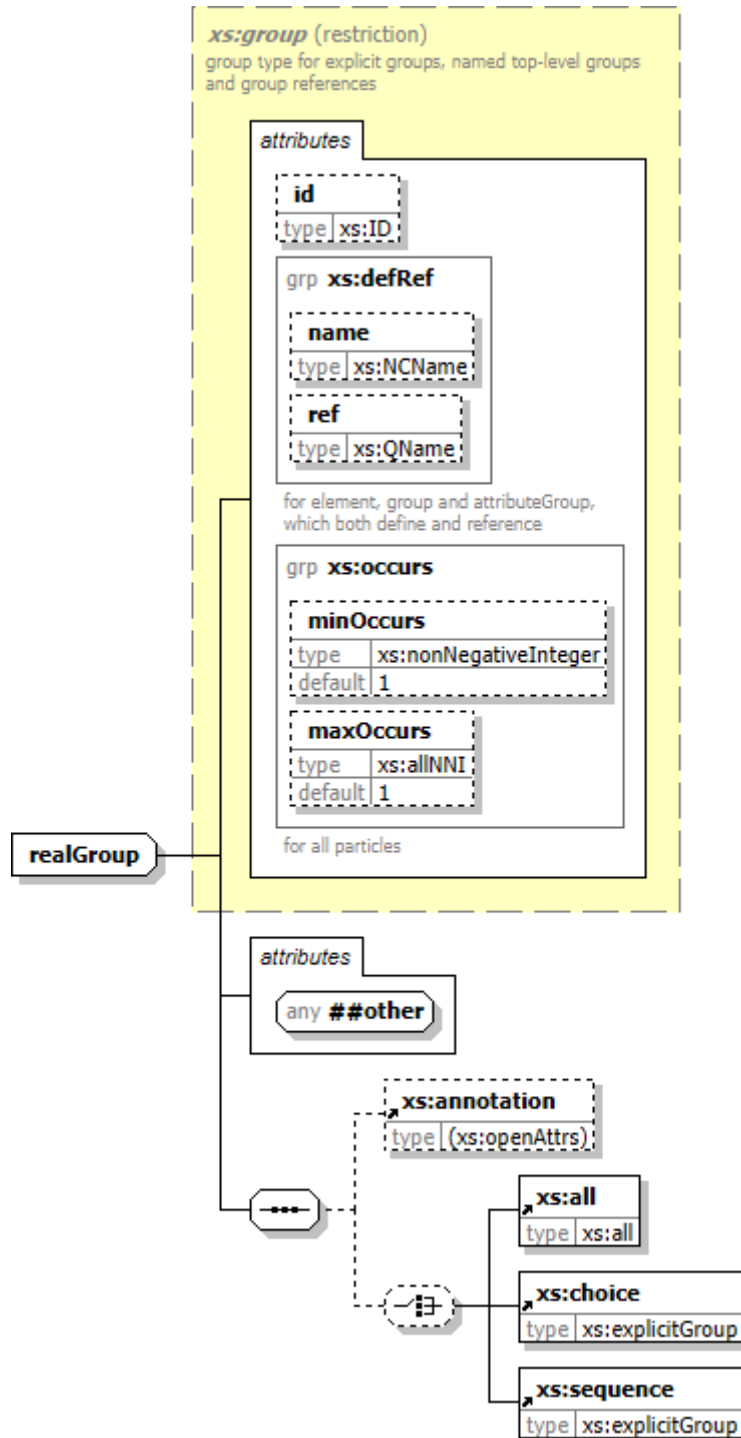
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 5 attributes, attr. wildcard, 4 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [227]

Component Diagram



XML Representation Summary

```
<...
  id           = xs:ID
  name        = xs:NCName
  ref         = xs:QName
  minOccurs   = xs:nonNegativeInteger : "1"
  maxOccurs   = (xs:nonNegativeInteger | "unbounded") : "1"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:all | xs:choice | xs:sequence)?
</...>
```

Content Model Elements (4):

[xs:all](#) [12], [xs:annotation](#) [17], [xs:choice](#) [36], [xs:sequence](#) [128]

Known Direct Subtypes (2):

[xs:groupRef](#) [192], [xs:namedGroup](#) [212]

All Direct / Indirect Based Elements (2):

[xs:group](#) [77], [xs:group](#) (type [xs:groupRef](#)) [79]

Known Usage Locations

- In derivations of other global types (2):

[xs:groupRef](#) [192] (as restriction base), [xs:namedGroup](#) [212] (as restriction base)

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:group [188] (restriction)
│           └── xs:realGroup
```

XML Source (see within schema source: p. 405)

```
<xs:complexType name="realGroup">
  <xs:complexContent>
    <xs:restriction base="xs:group">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice maxOccurs="1" minOccurs="0">
          <xs:element ref="xs:all"/>
          <xs:element ref="xs:choice"/>
          <xs:element ref="xs:sequence"/>
        </xs:choice>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations: 6/6)

■ id

Type: [xs:ID](#) [295]
Use: optional

Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Type: [xs:allNNI](#) [263]
 Use: optional
 Defined: [locally](#) [365] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

`xs:nonNegativeInteger | "unbounded"`

Default: "1"

minOccurs

Type: [xs:nonNegativeInteger](#) [314]
 Use: optional
 Defined: [locally](#) [366] within [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

name

Type: [xs:NCName](#) [308]
 Use: optional
 Defined: [locally](#) [363] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

ref

Type: [xs:QName](#) [322]
 Use: optional
 Defined: [locally](#) [364] within [xs:defRef](#) attributeGroup; see [XML source](#) [400]

{any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:realGroup](#) complexType; see [XML source](#) [406]

Content Element Detail (all declarations; 4/4)

[xs:all](#) [12]

Type: [xs:all](#) [150], complex content
 Defined: by reference within ([this](#)) [xs:realGroup](#) complexType; see [XML source](#) [406]

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
 Defined: by reference within ([this](#)) [xs:realGroup](#) complexType; see [XML source](#) [406]

[xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content
 Defined: by reference within ([this](#)) [xs:realGroup](#) complexType; see [XML source](#) [406]

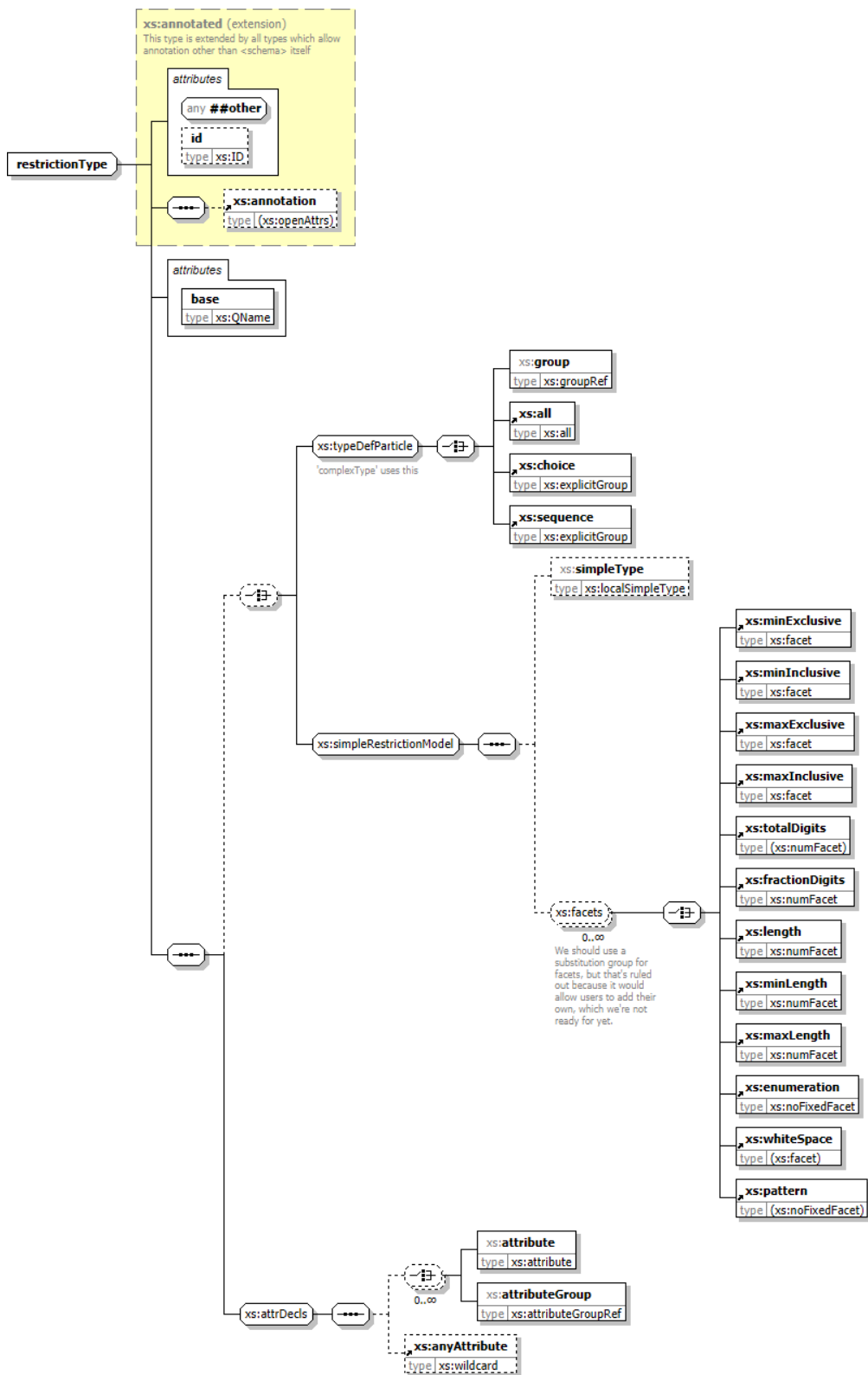
[xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content
 Defined: by reference within ([this](#)) [xs:realGroup](#) complexType; see [XML source](#) [406]

complexType xs:restrictionType

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 2 attributes, attr. wildcard, 21 elements
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [230]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:group | xs:all | xs:choice | xs:sequence | (xs:simpleType?,
  (xs:minExclusive | xs:minInclusive | xs:maxExclusive | xs:maxInclusive | xs:totalDigits |
  xs:fractionDigits | xs:length | xs:minLength | xs:maxLength | xs:enumeration | xs:whiteSpace
  | xs:pattern)*))?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?
</...>
```

Content Model Elements (21):

xs:all [12],	xs:maxInclusive [96],
xs:annotation [17],	xs:maxLength [98],
xs:anyAttribute [23],	xs:minExclusive [100],
xs:attribute (type xs:attribute) [29],	xs:minInclusive [102],
xs:attributeGroup (type xs:attributeGroupRef) [34],	xs:minLength [104],
xs:choice [36],	xs:pattern [109],
xs:enumeration [65],	xs:sequence [128],
xs:fractionDigits [75],	xs:simpleType (type xs:localSimpleType) [138],
xs:group (type xs:groupRef) [79],	xs:totalDigits [140],
xs:length [90],	xs:whiteSpace [147]
xs:maxExclusive [94],	

Known Direct Subtypes (2):

[xs:complexRestrictionType](#) [167], [xs:simpleRestrictionType](#) [239]

All Direct / Indirect Based Elements (2):

[xs:restriction](#) (in [xs:complexContent](#)) [118], [xs:restriction](#) (in [xs:simpleContent](#)) [121]

Known Usage Locations

- In derivations of other global types (2):

[xs:complexRestrictionType](#) [167] (as restriction base),
[xs:simpleRestrictionType](#) [239] (as restriction base)

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├─ xs:openAttrs [224] (extension)
│   └─ xs:annotated [153] (extension)
│       └─ xs:restrictionType
```

XML Source (see within schema source: p. 402)

```
<xs:complexType name="restrictionType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:group ref="xs:typeDefParticle" />
          <xs:group ref="xs:simpleRestrictionModel" />
        </xs:choice>
        <xs:group ref="xs:attrDecls" />
      </xs:sequence>
      <xs:attribute name="base" type="xs:QName" use="required" />
    </xs:extension>
```

</xs:complexContent>
</xs:complexType>

Attribute Detail (all declarations; 3/3)

- base

Type: [xs:QName](#) [322]
Use: required
Defined: locally within (this) [xs:restrictionType](#) complexType; see [XML source](#) [402]
- id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]
- {any attribute from non-schema namespace}

Defined: locally [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 21/21)

- [xs:all](#) [12]

Type: [xs:all](#) [150], complex content
Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]
- [xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference [155] within [xs:annotated](#) complexType; see [XML source](#) [397]
- [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: by reference [340] within [xs:attrDecls](#) group; see [XML source](#) [401]
- [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: locally [340] within [xs:attrDecls](#) group; see [XML source](#) [401]
- [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: locally [341] within [xs:attrDecls](#) group; see [XML source](#) [401]
- [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: by reference [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]
- [xs:enumeration](#) [65]

Type: [xs:noFixedFacet](#) [220], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]
- [xs:fractionDigits](#) [75]

Type: [xs:numFacet](#) [222], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]
- [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: [locally](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:length](#) [90]

Type: [xs:numFacet](#) [222], complex content

Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxExclusive](#) [94]

Type: [xs:facet](#) [186], complex content

Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxInclusive](#) [96]

Type: [xs:facet](#) [186], complex content

Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxLength](#) [98]

Type: [xs:numFacet](#) [222], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minExclusive](#) [100]

Type: [xs:facet](#) [186], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minInclusive](#) [102]

Type: [xs:facet](#) [186], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minLength](#) [104]

Type: [xs:numFacet](#) [222], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:pattern](#) [109]

Type: [anonymous complexType \(restriction of xs:noFixedFacet\)](#) [110], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content

Defined: [locally](#) [360] within [xs:simpleRestrictionModel](#) group; see [XML source](#) [424]

↔ [xs:totalDigits](#) [140]

Type: [anonymous complexType \(restriction of xs:numFacet\)](#) [141], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:whiteSpace](#) [147]

Type: [anonymous complexType \(restriction of xs:facet\)](#) [148], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

complexType xs:simpleExplicitGroup

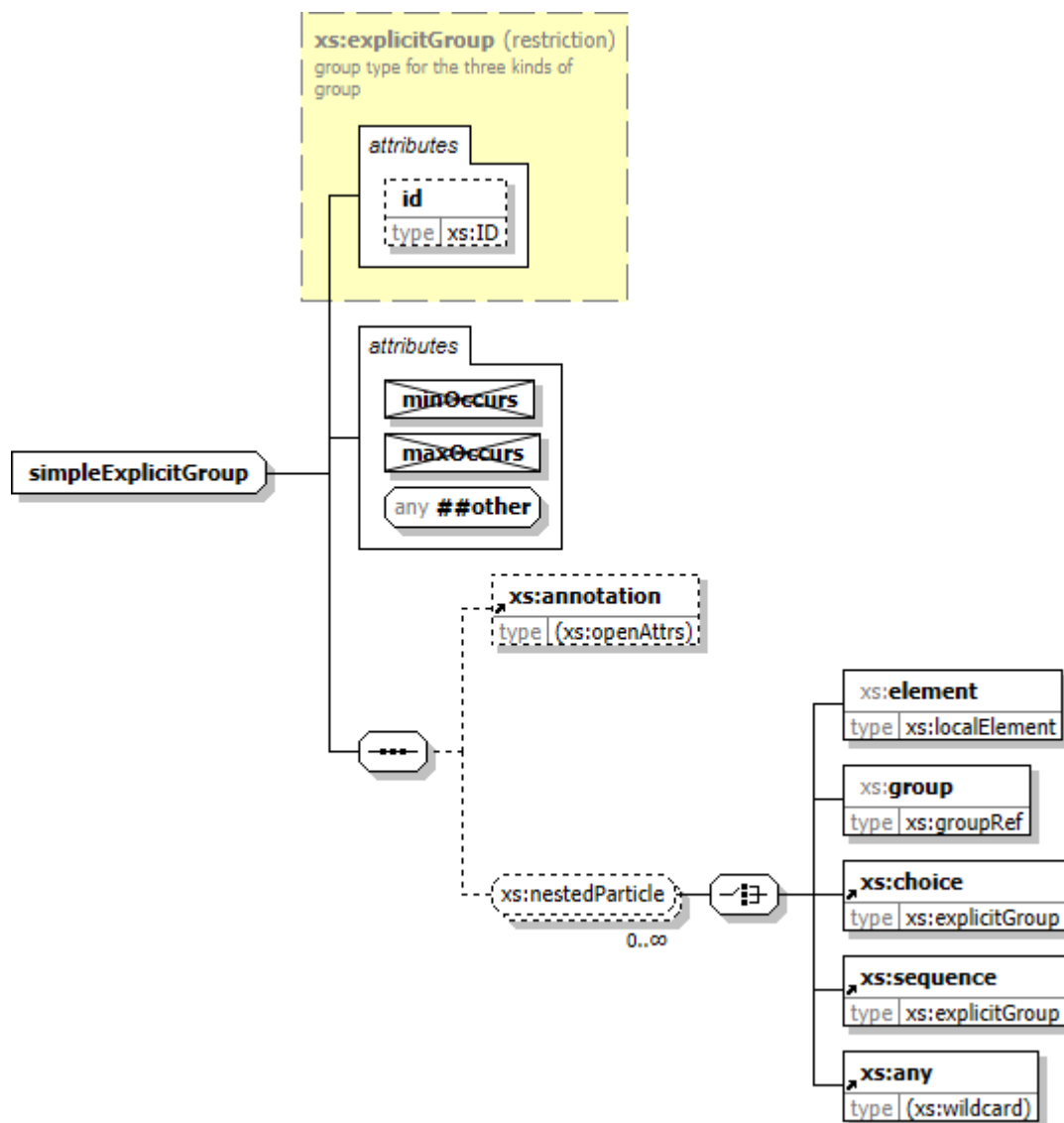
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 1 attribute, attr. wildcard, 6 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [234]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:element | xs:group | xs:choice | xs:sequence | xs:any)*
</...>

```

Content Model Elements (6):

[xs:annotation](#) [17], [xs:element](#) (type [xs:localElement](#)) [57],
[xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]

All Direct / Indirect Based Elements (2):

[xs:choice](#) (in [xs:group](#)) [39], [xs:sequence](#) (in [xs:group](#)) [131]

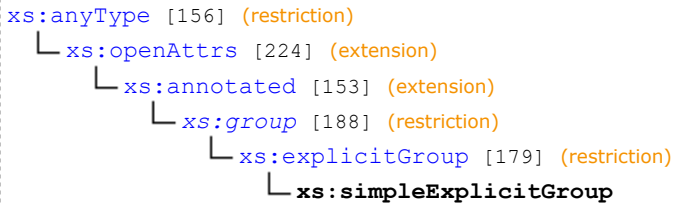
Known Usage Locations

- As direct type of elements (2):

[xs:choice](#) (in [xs:group](#)) [39], [xs:sequence](#) (in [xs:group](#)) [131]

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 406)

```

<xs:complexType name="simpleExplicitGroup">
  <xs:complexContent>
    <xs:restriction base="xs:explicitGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:nestedParticle"/>
      </xs:sequence>
      <xs:attribute name="minOccurs" use="prohibited"/>
      <xs:attribute name="maxOccurs" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

id

Type: [xs:ID](#) [295]
 Use: optional
 Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

maxOccurs

Use: prohibited

minOccurs

Use: prohibited

{any attribute from non-schema namespace}

Defined: locally within (this) [xs:simpleExplicitGroup](#) complexType; see [XML source](#) [407]

Content Element Detail (all declarations; 6/6)

[xs:annotation](#) [17]

Type: anonymous complexType (extension of [xs:openAttrs](#)) [18], complex content
 Defined: by reference within (this) [xs:simpleExplicitGroup](#) complexType; see [XML source](#) [407]

↔ [xs:any](#) [20]

Type: [anonymous complexType \(extension of xs:wildcard\)](#) [21], complex content
Defined: [by reference](#) [349] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content
Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content
Defined: [locally](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

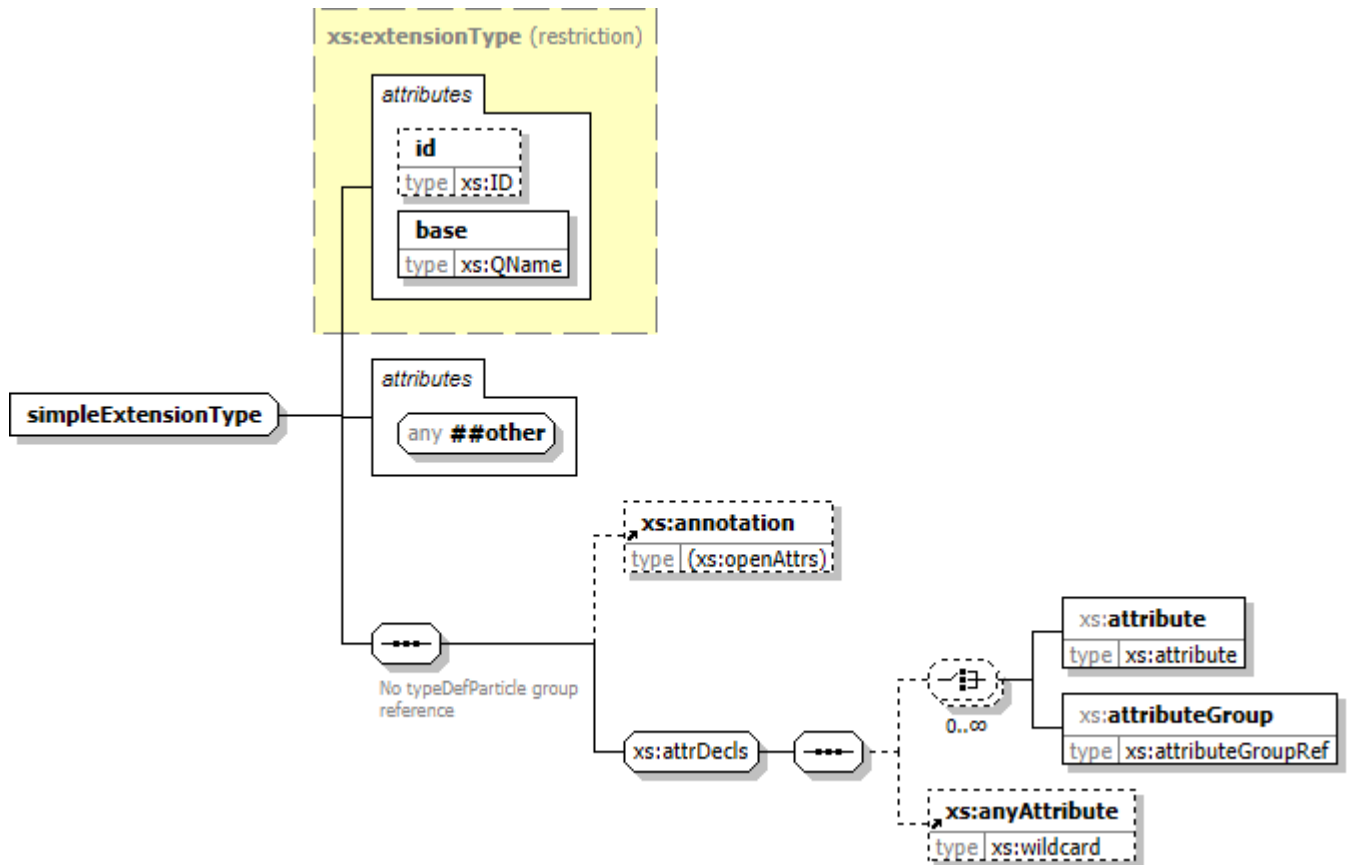
↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [350] within [xs:nestedParticle](#) group; see [XML source](#) [400]

complexType xs:simpleExtensionType

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 2 attributes, attr. wildcard, 4 elements
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [237]

Component Diagram



XML Representation Summary

```

<...
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?
</...>
    
```

Content Model Elements (4):

- [xs:annotation](#) [17], [xs:attribute](#) (type [xs:attribute](#)) [29],
- [xs:anyAttribute](#) [23], [xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

All Direct / Indirect Based Elements (1):

- [xs:extension](#) (in [xs:simpleContent](#)) [70]

Known Usage Locations

- As direct type of elements (1):
 - [xs:extension](#) (in [xs:simpleContent](#)) [70]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:extensionType [183] (restriction)
│           └── xs:simpleExtensionType
    
```

XML Source (see within schema source: p. 403)

```

<xs:complexType name="simpleExtensionType">
  <xs:complexContent>
    <xs:restriction base="xs:extensionType">
      <xs:sequence>
        <xs:annotation>
          <xs:documentation>
            No typeDefParticle group reference
          </xs:documentation>
        </xs:annotation>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:attrDecls"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 3/3)

base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [184] within [xs:extensionType](#) complexType; see [XML source](#) [403]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:simpleExtensionType](#) complexType; see [XML source](#) [404]

Content Element Detail (all declarations; 4/4)

↔ [xs:annotation](#) [17]


Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:simpleExtensionType](#) complexType; see [XML source](#) [404]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: by reference [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ [xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

 [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content

Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

complexType xs:simpleRestrictionType

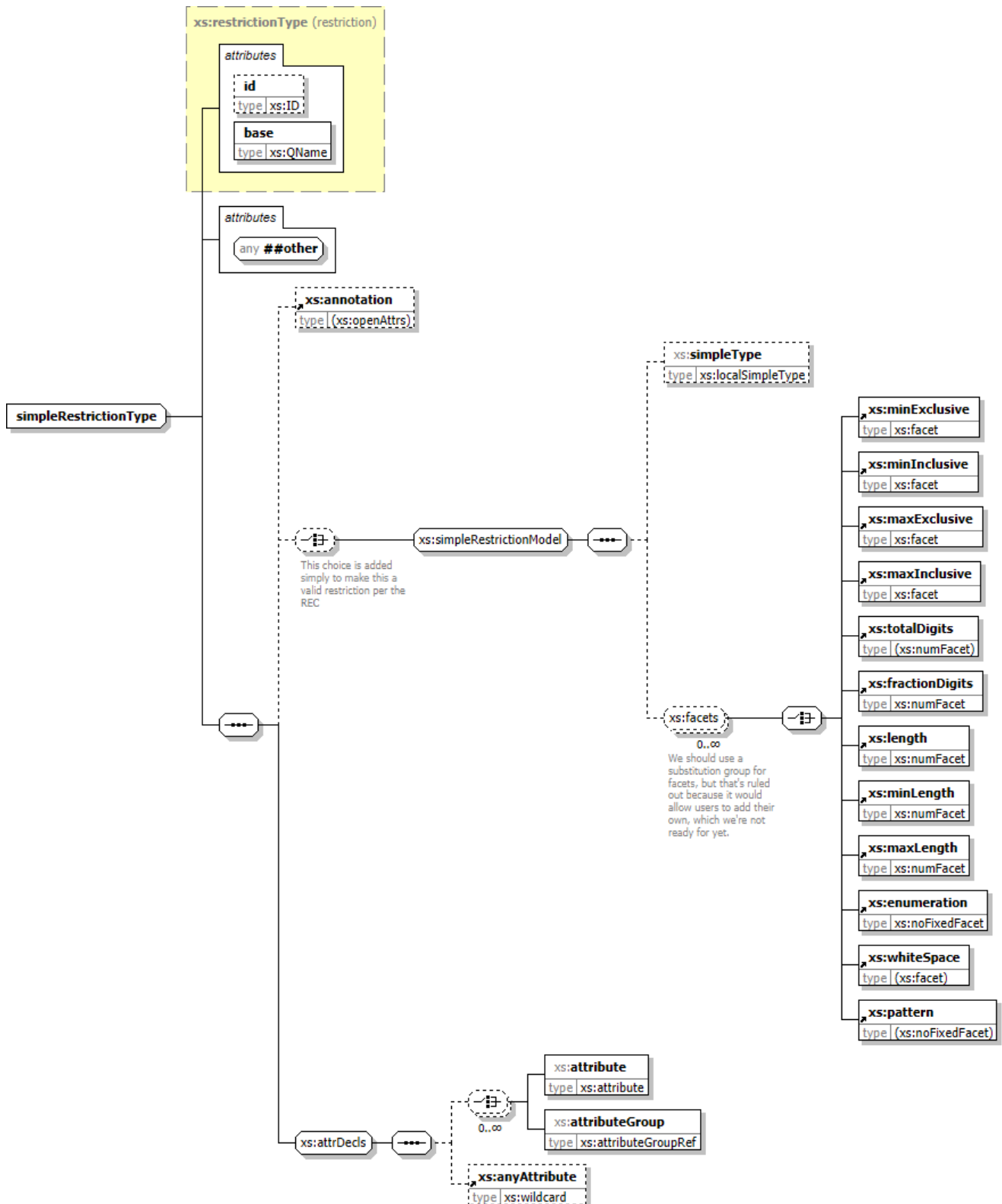
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 2 attributes, attr. wildcard, 17 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [240]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  base = xs:QName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType?, (xs:minExclusive | xs:minInclusive | xs:maxExclusive |
xs:maxInclusive | xs:totalDigits | xs:fractionDigits | xs:length | xs:minLength | xs:maxLength |
xs:enumeration | xs:whiteSpace | xs:pattern)*)?, (xs:attribute | xs:attributeGroup)*,
xs:anyAttribute?
</...>
```

Content Model Elements (17):

xs:annotation [17],	xs:maxLength [98],
xs:anyAttribute [23],	xs:minExclusive [100],
xs:attribute (type xs:attribute) [29],	xs:minInclusive [102],
xs:attributeGroup (type xs:attributeGroupRef) [34],	xs:minLength [104],
xs:enumeration [65],	xs:pattern [109],
xs:fractionDigits [75],	xs:simpleType (type xs:localSimpleType) [138],
xs:length [90],	xs:totalDigits [140],
xs:maxExclusive [94],	xs:whiteSpace [147]
xs:maxInclusive [96],	

All Direct / Indirect Based Elements (1):

xs:restriction (in [xs:simpleContent](#)) [121]

Known Usage Locations

- As direct type of elements (1):

xs:restriction (in [xs:simpleContent](#)) [121]

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:restrictionType [229] (restriction)
│           └── xs:simpleRestrictionType
```

XML Source (see within schema source: p. 403)

```
<xs:complexType name="simpleRestrictionType">
  <xs:complexContent>
    <xs:restriction base="xs:restrictionType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation" />
        <xs:choice minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              This choice is added simply to
              make this a valid restriction per the REC
            </xs:documentation>
          </xs:annotation>
          <xs:group ref="xs:simpleRestrictionModel" />
        </xs:choice>
        <xs:group ref="xs:attrDecls" />
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:restriction>
  </xs:complexContent>
```

</xs:complexType>

Attribute Detail (all declarations; 3/3)

base

Type: [xs:QName](#) [322]
Use: required
Defined: [locally](#) [231] within [xs:restrictionType](#) complexType; see [XML source](#) [402]

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

{any attribute from non-schema namespace}

Defined: locally within (this) [xs:simpleRestrictionType](#) complexType; see [XML source](#) [403]

Content Element Detail (all declarations; 17/17)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType (extension of [xs:openAttrs](#)) [18], complex content
Defined: by reference within (this) [xs:simpleRestrictionType](#) complexType; see [XML source](#) [403]

[xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
Defined: by reference [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:enumeration](#) [65]

Type: [xs:noFixedFacet](#) [220], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]

[xs:fractionDigits](#) [75]

Type: [xs:numFacet](#) [222], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]

[xs:length](#) [90]

Type: [xs:numFacet](#) [222], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]

[xs:maxExclusive](#) [94]

Type: [xs:facet](#) [186], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]

[xs:maxInclusive](#) [96]

Type: [xs:facet](#) [186], complex content
Defined: by reference [345] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxLength](#) [98]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minExclusive](#) [100]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minInclusive](#) [102]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minLength](#) [104]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:pattern](#) [109]

Type: [anonymous complexType \(restriction of xs:noFixedFacet\)](#) [110], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: [locally](#) [360] within [xs:simpleRestrictionModel](#) group; see [XML source](#) [424]

↔ [xs:totalDigits](#) [140]

Type: [anonymous complexType \(restriction of xs:numFacet\)](#) [141], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

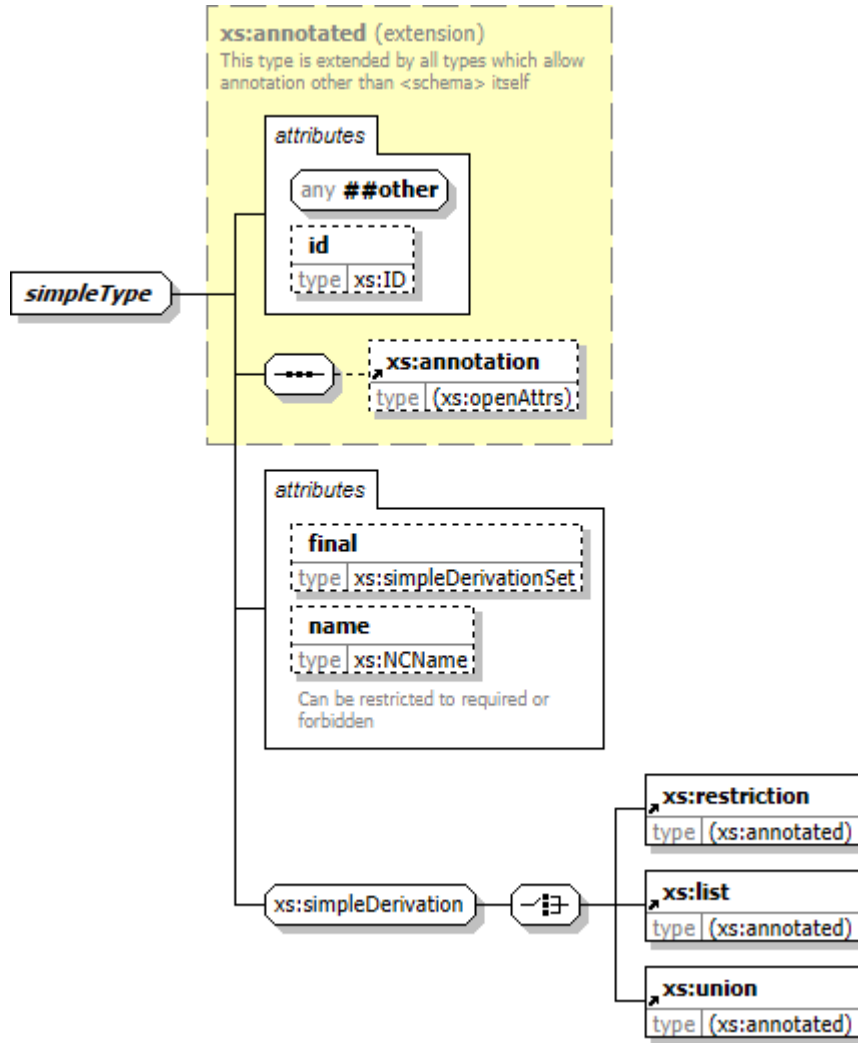
↔ [xs:whiteSpace](#) [147]

Type: [anonymous complexType \(restriction of xs:facet\)](#) [148], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

complexType xs:simpleType

Namespace: <http://www.w3.org/2001/XMLSchema>
Content: complex, 3 attributes, attr. wildcard, 4 elements
Abstract: (cannot be assigned directly to elements used in instance XML documents)
Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [244]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  final = ("#all" | list of ("list" | "union" | "restriction"))
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:restriction | xs:list | xs:union)
</...>
```

Content Model Elements (4):

[xs:annotation](#) [17], [xs:list](#) [92], [xs:restriction](#) [114], [xs:union](#) [142]

Known Direct Subtypes (2):

[xs:localSimpleType](#) [206], [xs:topLevelSimpleType](#) [257]

All Direct / Indirect Based Elements (2):

[xs:simpleType](#) [135], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

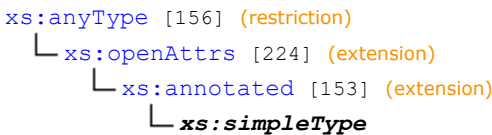
Known Usage Locations

- **In derivations of other global types (2):**

[xs:localSimpleType](#) [206] (as restriction base), [xs:topLevelSimpleType](#) [257] (as restriction base)

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 423)

```

<xs:complexType abstract="true" name="simpleType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group ref="xs:simpleDerivation"/>
      <xs:attribute name="final" type="xs:simpleDerivationSet"/>
      <xs:attribute name="name" type="xs:NCName">
        <xs:annotation>
          <xs:documentation>
            Can be restricted to required or forbidden
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

■ final

Type: [xs:simpleDerivationSet](#) [326]
Use: optional
Defined: locally within (this) [xs:simpleType](#) complexType; see [XML source](#) [423]

Attribute Value

"#all" | list of ("list" | "union" | "restriction")

■ id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ name

Type: [xs:NCName](#) [308]
Use: optional
Defined: locally within (this) [xs:simpleType](#) complexType; see [XML source](#) [423]

Can be restricted to required or forbidden

■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within [xs:openAttrs](#) complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 4/4)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: [by reference](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

↔ [xs:list](#) [92]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [93], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

↔ [xs:restriction](#) [114]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [115], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

↔ [xs:union](#) [142]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [143], complex content
Defined: [by reference](#) [357] within [xs:simpleDerivation](#) group; see [XML source](#) [423]

complexType xs:topLevelAttribute

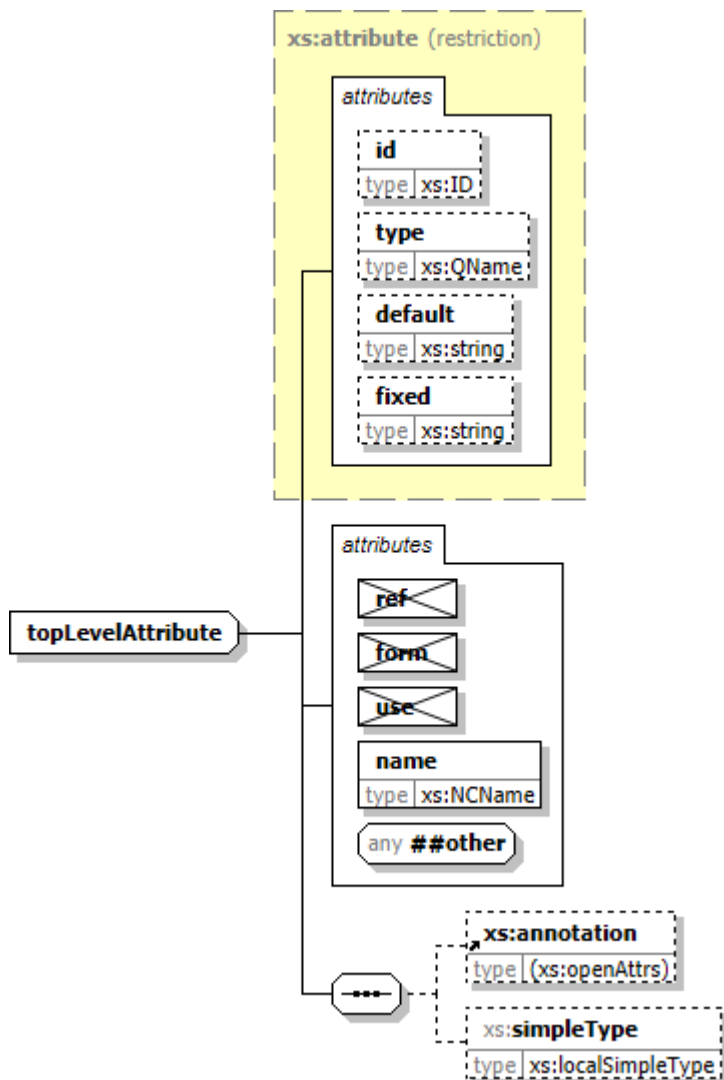
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 5 attributes, attr. wildcard, 2 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [247]

Component Diagram



XML Representation Summary

```

<...
  id       = xs:ID
  type     = xs:QName
  default  = xs:string
  fixed    = xs:string
  name     = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, xs:simpleType?
</...>

```

Content Model Elements (2):

[xs:annotation](#) [17], [xs:simpleType](#) (type [xs:localSimpleType](#)) [138]

All Direct / Indirect Based Elements (1):

[xs:attribute](#) [27]

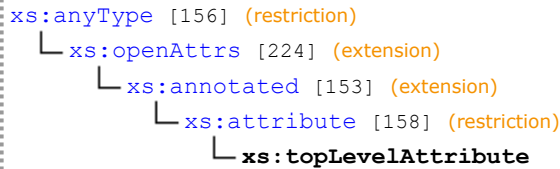
Known Usage Locations

- As direct type of elements (1):

[xs:attribute](#) [27]

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 401)

```

<xs:complexType name="topLevelAttribute">
  <xs:complexContent>
    <xs:restriction base="xs:attribute">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
      </xs:sequence>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:attribute name="form" use="prohibited"/>
      <xs:attribute name="use" use="prohibited"/>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 9/9)

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

form

Use: prohibited

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]

Use: required
Defined: locally within ([this](#)) [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

■ ~~ref~~

Use: prohibited

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [160] within [xs:attribute](#) complexType; see [XML source](#) [401]

■ ~~use~~

Use: prohibited

■ {any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

Content Element Detail (all declarations; 2/2)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: locally within ([this](#)) [xs:topLevelAttribute](#) complexType; see [XML source](#) [401]

complexType

xs:topLevelComplexType

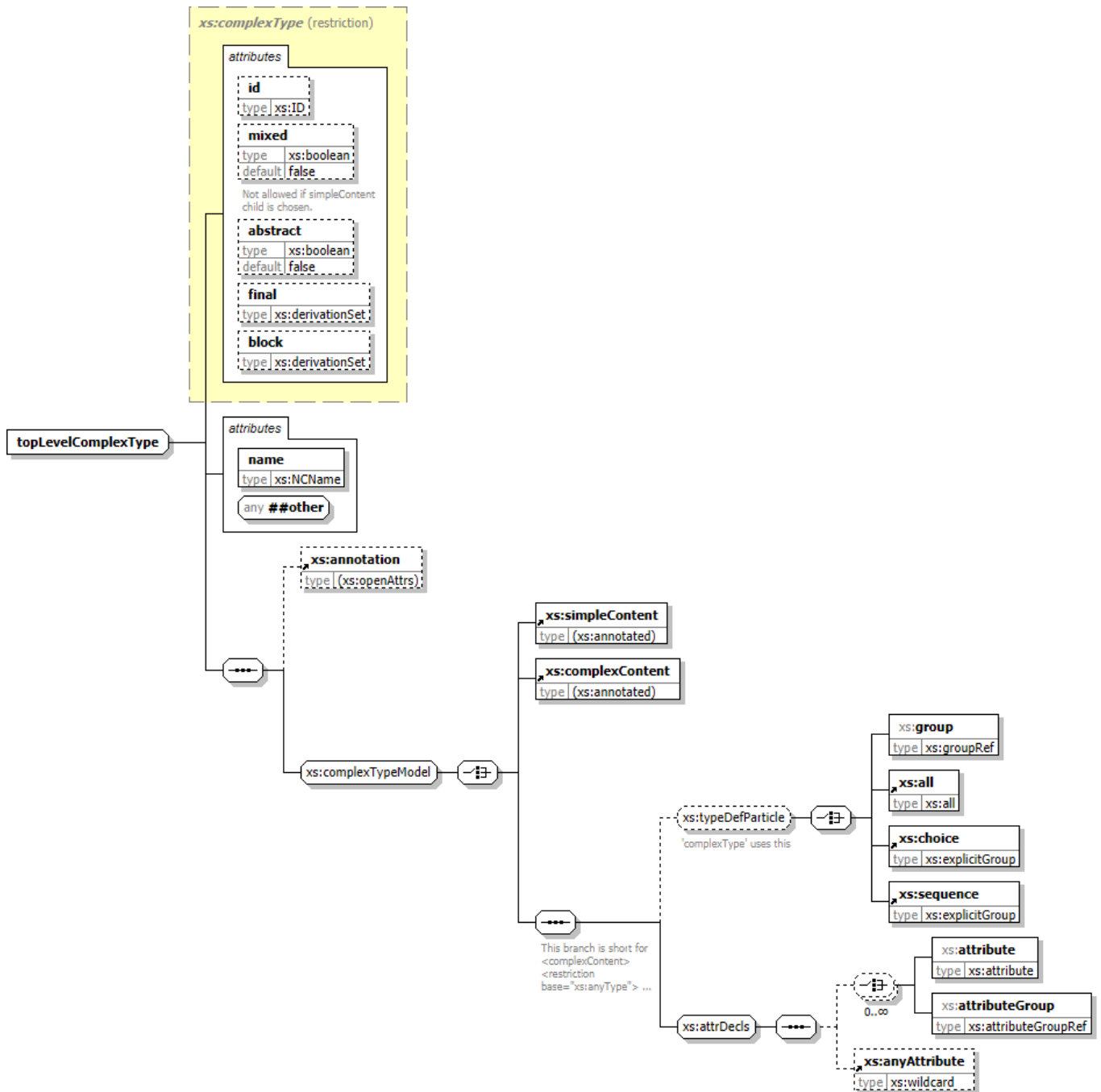
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 6 attributes, attr. wildcard, 10 elements

Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [250]

Component Diagram



XML Representation Summary

```
<...
  id           = xs:ID
  mixed       = xs:boolean : "false"
  abstract    = xs:boolean : "false"
  final       = ("#all" | list of ("extension" | "restriction"))
  block       = ("#all" | list of ("extension" | "restriction"))
  name        = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleContent | xs:complexContent | ((xs:group | xs:all | xs:choice |
  xs:sequence)?, (xs:attribute | xs:attributeGroup)*, xs:anyAttribute?))
</...>
```

Content Model Elements (10):

xs:all [12],	xs:choice [36],
xs:annotation [17],	xs:complexContent [41],
xs:anyAttribute [23],	xs:group (type xs:groupRef) [79],
xs:attribute (type xs:attribute) [29],	xs:sequence [128],
xs:attributeGroup (type xs:attributeGroupRef) [34],	xs:simpleContent [133]

All Direct / Indirect Based Elements (1):

[xs:complexType](#) [44]

Known Usage Locations

- As direct type of elements (1):

[xs:complexType](#) [44]

Type Definition Detail

Type Derivation Tree

```
xs:anyType [156] (restriction)
├─ xs:openAttrs [224] (extension)
│   └─ xs:annotated [153] (extension)
│       └─ xs:complexType [170] (restriction)
│           └─ xs:topLevelComplexType
```

XML Source (see within schema source: p. 402)

```
<xs:complexType name="topLevelComplexType">
  <xs:complexContent>
    <xs:restriction base="xs:complexType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation" />
        <xs:group ref="xs:complexTypeModel" />
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required" />
      <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 7/7)

abstract

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Attribute Value

Default: "false"

■ block

Type: [xs:derivationSet](#) [279]
 Use: optional
 Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

■ final

Type: [xs:derivationSet](#) [279]
 Use: optional
 Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

■ id

Type: [xs:ID](#) [295]
 Use: optional
 Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

■ mixed

Type: [xs:boolean](#) [270]
 Use: optional
 Defined: [locally](#) [172] within [xs:complexType](#) complexType; see [XML source](#) [402]

Not allowed if simpleContent child is chosen.
 May be overridden by setting on complexContent child.

Attribute Value

Default: "false"

■ name

Type: [xs:NCName](#) [308]
 Use: required
 Defined: [locally](#) within ([this](#)) [xs:topLevelComplexType](#) complexType; see [XML source](#) [402]

■ {any attribute from non-schema namespace}

Defined: [locally](#) within ([this](#)) [xs:topLevelComplexType](#) complexType; see [XML source](#) [402]

Content Element Detail (all declarations; 10/10)

↔ [xs:all](#) [12]

Type: [xs:all](#) [150], complex content
 Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of](#) [xs:openAttrs](#)) [18], complex content
 Defined: [by reference](#) within ([this](#)) [xs:topLevelComplexType](#) complexType; see [XML source](#) [402]

↔ [xs:anyAttribute](#) [23]

Type: [xs:wildcard](#) [260], complex content
 Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ `xs:attribute` [29]

Type: `xs:attribute` [158], complex content
Defined: `locally` [340] within `xs:attrDecls` group; see [XML source](#) [401]

↔ `xs:attributeGroup` [34]

Type: `xs:attributeGroupRef` [165], complex content
Defined: `locally` [341] within `xs:attrDecls` group; see [XML source](#) [401]

↔ `xs:choice` [36]

Type: `xs:explicitGroup` [179], complex content
Defined: `by reference` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:complexContent` [41]

Type: `anonymous complexType (extension of xs:annotated)` [42], complex content
Defined: `by reference` [343] within `xs:complexTypeModel` group; see [XML source](#) [401]

↔ `xs:group` [79]

Type: `xs:groupRef` [192], complex content
Defined: `locally` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:sequence` [128]

Type: `xs:explicitGroup` [179], complex content
Defined: `by reference` [362] within `xs:typeDefParticle` group; see [XML source](#) [400]

↔ `xs:simpleContent` [133]

Type: `anonymous complexType (extension of xs:annotated)` [134], complex content
Defined: `by reference` [343] within `xs:complexTypeModel` group; see [XML source](#) [401]

complexType xs:topLevelElement

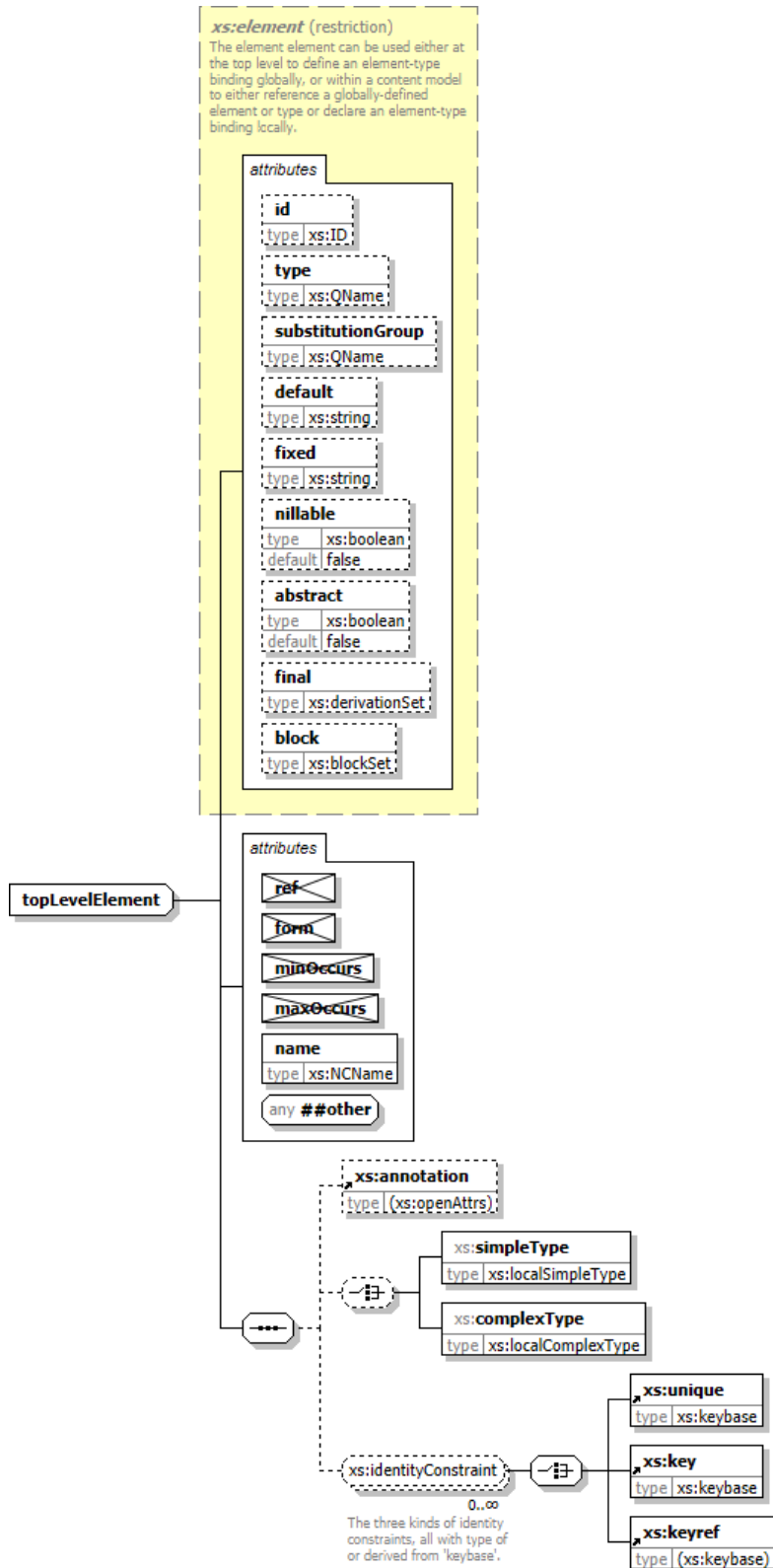
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 10 [attributes](#), attr. [wildcard](#), 6 [elements](#)

Block: "#all" (*blocks all substitutions of this complex type through xsi:type attribute in instance XML documents*)

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [254]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  type = xs:QName
  substitutionGroup = xs:QName
  default = xs:string
  fixed = xs:string
  nillable = xs:boolean : "false"
  abstract = xs:boolean : "false"
  final = ("#all" | list of ("extension" | "restriction"))
  block = ("#all" | list of ("extension" | "restriction" | "substitution"))
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:simpleType | xs:complexType)?, (xs:unique | xs:key | xs:keyref)*
</...>
```

Content Model Elements (6):

- [xs:annotation](#) [17],
- [xs:complexType](#) (type [xs:localComplexType](#)) [48],
- [xs:key](#) [85],
- [xs:keyref](#) [87],
- [xs:simpleType](#) (type [xs:localSimpleType](#)) [138],
- [xs:unique](#) [145]

All Direct / Indirect Based Elements (1):

- [xs:element](#) [53]

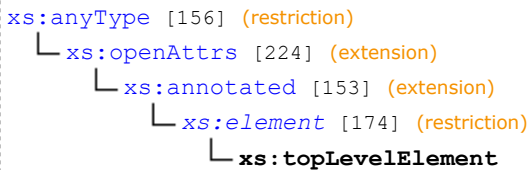
Known Usage Locations

- As direct type of elements (1):

- [xs:element](#) [53]

Type Definition Detail

Type Derivation Tree



XML Source (see within schema source: p. 405)

```
<xs:complexType name="topLevelElement">
  <xs:complexContent>
    <xs:restriction base="xs:element">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:attribute name="form" use="prohibited"/>
      <xs:attribute name="minOccurs" use="prohibited"/>
      <xs:attribute name="maxOccurs" use="prohibited"/>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

Attribute Detail (all declarations; 15/15)

abstract

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

block

Type: [xs:blockSet](#) [268]
Use: optional
Defined: [locally](#) [176] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction" | "substitution")
```

default

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

final

Type: [xs:derivationSet](#) [279]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

```
"#all" | list of ("extension" | "restriction")
```

fixed

Type: [xs:string](#) [328]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

~~form~~

Use: prohibited

id

Type: [xs:ID](#) [295]
Use: optional
Defined: [locally](#) [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

~~maxOccurs~~

Use: prohibited

~~minOccurs~~

Use: prohibited

name

Type: [xs:NCName](#) [308]
Use: required
Defined: locally within ([this](#)) [xs:topLevelElement](#) complexType; see [XML source](#) [405]

■ nillable

Type: [xs:boolean](#) [270]
Use: optional
Defined: [locally](#) [177] within [xs:element](#) complexType; see [XML source](#) [405]

Attribute Value

Default: "false"

■ ref

Use: prohibited

■ substitutionGroup

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ type

Type: [xs:QName](#) [322]
Use: optional
Defined: [locally](#) [178] within [xs:element](#) complexType; see [XML source](#) [405]

■ {any attribute from non-schema namespace}

Defined: locally within ([this](#)) [xs:topLevelElement](#) complexType; see [XML source](#) [405]

Content Element Detail (all declarations; 6/6)

↔ [xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:topLevelElement](#) complexType; see [XML source](#) [405]

↔ [xs:complexType](#) [48]

Type: [xs:localComplexType](#) [197], complex content
Defined: locally within ([this](#)) [xs:topLevelElement](#) complexType; see [XML source](#) [405]

↔ [xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [347] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:keyref](#) [87]

Type: [anonymous](#) complexType ([extension of xs:keybase](#)) [88], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content
Defined: locally within ([this](#)) [xs:topLevelElement](#) complexType; see [XML source](#) [405]

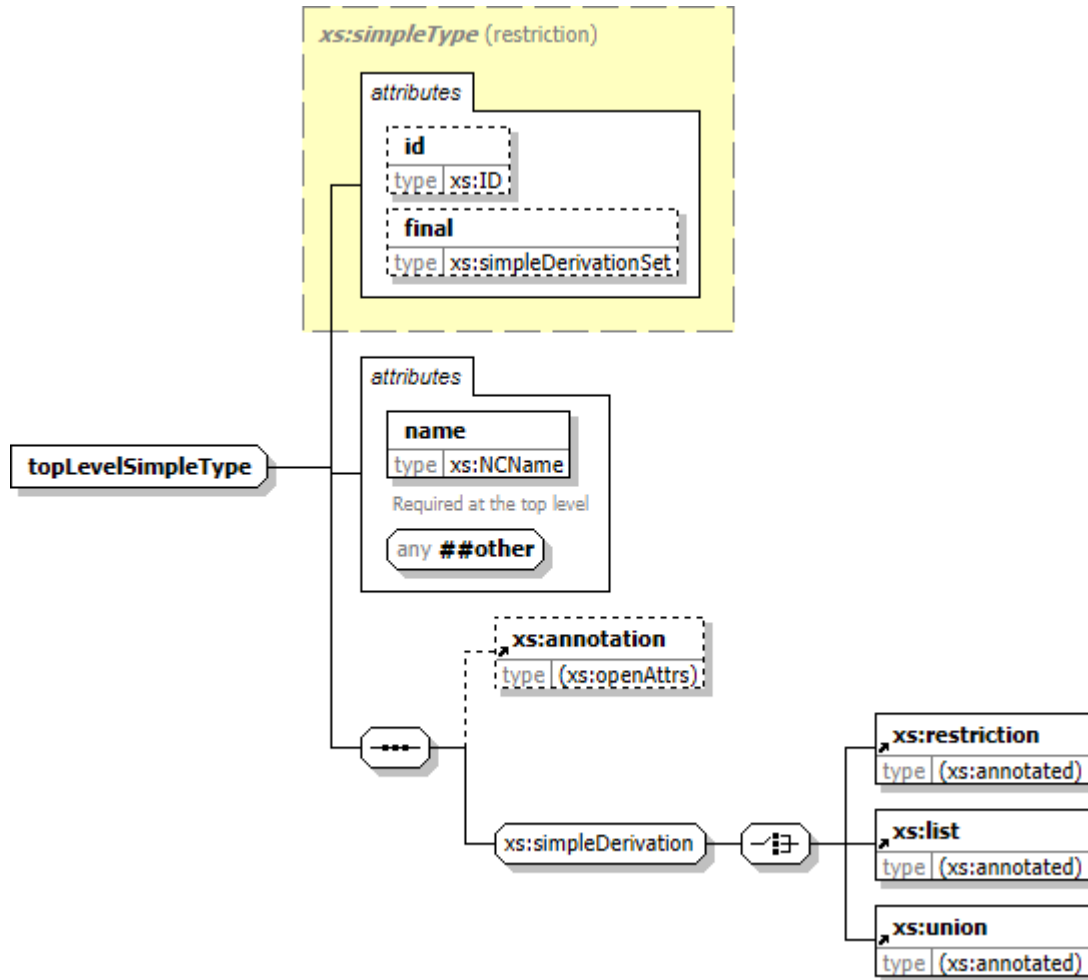
↔ [xs:unique](#) [145]

Type: [xs:keybase](#) [195], complex content
Defined: by reference [348] within [xs:identityConstraint](#) group; see [XML source](#) [411]

complexType xs:topLevelSimpleType

Namespace: <http://www.w3.org/2001/XMLSchema>
 Content: complex, 3 attributes, attr. wildcard, 4 elements
 Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [258]

Component Diagram



XML Representation Summary

```
<...
  id = xs:ID
  final = ("#all" | list of ("list" | "union" | "restriction"))
  name = xs:NCName
  {any attribute from non-schema namespace}
>
Content: xs:annotation?, (xs:restriction | xs:list | xs:union)
</...>
```

Content Model Elements (4):

[xs:annotation](#) [17], [xs:list](#) [92], [xs:restriction](#) [114], [xs:union](#) [142]

All Direct / Indirect Based Elements (1):

[xs:simpleType](#) [135]

Known Usage Locations

- As direct type of elements (1):

[xs:simpleType](#) [135]

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:simpleType [243] (restriction)
│           └── xs:topLevelSimpleType
    
```

XML Source (see within schema source: p. 423)

```

<xs:complexType name="topLevelSimpleType">
  <xs:complexContent>
    <xs:restriction base="xs:simpleType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:simpleDerivation"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required">
        <xs:annotation>
          <xs:documentation>
            Required at the top level
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

final

Type: [xs:simpleDerivationSet](#) [326]
Use: optional
Defined: locally [244] within [xs:simpleType](#) complexType; see [XML source](#) [423]

Attribute Value

```
"#all" | list of ("list" | "union" | "restriction")
```

id

Type: [xs:ID](#) [295]
Use: optional
Defined: locally [155] within [xs:annotated](#) complexType; see [XML source](#) [397]

name

Type: [xs:NCName](#) [308]
Use: required
Defined: locally within (this) [xs:topLevelSimpleType](#) complexType; see [XML source](#) [424]

Required at the top level

{any attribute from non-schema namespace}

Defined: locally within (this) [xs:topLevelSimpleType](#) complexType; see [XML source](#) [424]

Content Element Detail (all declarations; 4/4)

`xs:annotation` [17]

Type: `anonymous` complexType (extension of `xs:openAttrs`) [18], complex content

Defined: by reference within (this) `xs:topLevelSimpleType` complexType; see [XML source](#) [423]

`xs:list` [92]

Type: `anonymous` complexType (extension of `xs:annotated`) [93], complex content

Defined: by reference [357] within `xs:simpleDerivation` group; see [XML source](#) [423]

`xs:restriction` [114]

Type: `anonymous` complexType (extension of `xs:annotated`) [115], complex content

Defined: by reference [357] within `xs:simpleDerivation` group; see [XML source](#) [423]

`xs:union` [142]

Type: `anonymous` complexType (extension of `xs:annotated`) [143], complex content

Defined: by reference [357] within `xs:simpleDerivation` group; see [XML source](#) [423]

complexType xs:wildcard

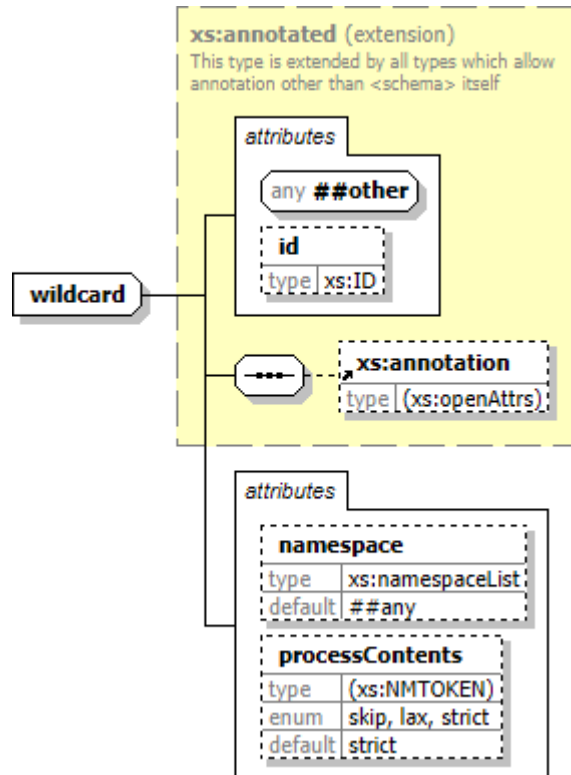
Namespace: <http://www.w3.org/2001/XMLSchema>

Content: complex, 3 attributes, attr. `wildcard`, 1 element

Block: "#all" (blocks all substitutions of this complex type through `xsi:type` attribute in instance XML documents)

Defined: globally in `XMLSchema.xsd`; see [XML source](#) [261]

Component Diagram



XML Representation Summary

```
<...
  id           = xs:ID
  namespace    = (("##any" | "##other") | list of (xs:anyURI | ("##targetNamespace" | "##local"))) :
                "##any"
  processContents = ("skip" | "lax" | "strict") : "strict"
  {any attribute from non-schema namespace}
>
Content: xs:annotation?
</...>
```

Content Model Elements (1):

[xs:annotation](#) [17]

All Direct / Indirect Based Elements (2):

[xs:any](#) [20], [xs:anyAttribute](#) [23]

Known Usage Locations

- As direct type of elements (1):

[xs:anyAttribute](#) [23]

- In derivations of anonymous types of elements (1):

`xs:any` [20] (as extension base)

Type Definition Detail

Type Derivation Tree

```

xs:anyType [156] (restriction)
├── xs:openAttrs [224] (extension)
│   └── xs:annotated [153] (extension)
│       └── xs:wildcard
    
```

XML Source (see within schema source: p. 408)

```

<xs:complexType name="wildcard">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:attribute default="##any" name="namespace" type="xs:namespaceList" use="optional"/>
      <xs:attribute default="strict" name="processContents" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="skip"/>
            <xs:enumeration value="lax"/>
            <xs:enumeration value="strict"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

Attribute Detail (all declarations; 4/4)

id

Type: `xs:ID` [295]
Use: optional
Defined: locally [155] within `xs:annotated` complexType; see [XML source](#) [397]

namespace

Type: `xs:namespaceList` [306]
Use: optional
Defined: locally within (this) `xs:wildcard` complexType; see [XML source](#) [408]

Attribute Value

```

( "##any" | "##other" ) | list of ( xs:anyURI | ( "##targetNamespace" | "##local" ) )
    
```

Default: "##any"

processContents

Type: anonymous simpleType (restriction of `xs:NMTOKEN`) [262]
Use: optional
Defined: locally within (this) `xs:wildcard` complexType; see [XML source](#) [408]

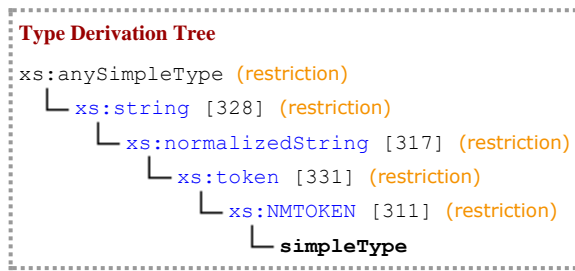
Attribute Value

```

enumeration of xs:NMTOKEN
    
```

Enumeration: "skip", "lax", "strict"
Default: "strict"

Anonymous simpleType



■ {any attribute from non-schema namespace}

Defined: [locally](#) [225] within `xs:openAttrs` complexType; see [XML source](#) [397]

Content Element Detail (all declarations; 1/1)

↔ `xs:annotation` [17]

Type: [anonymous](#) complexType (extension of `xs:openAttrs`) [18], complex content

Defined: [by reference](#) [155] within `xs:annotated` complexType; see [XML source](#) [397]

Simple Types

simpleType

xs:allNNI

Namespace: <http://www.w3.org/2001/XMLSchema>Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [263]

Simple Content Model

[xs:nonNegativeInteger](#) | "unbounded"

All Direct / Indirect Based Attributes (3):

[xs:all/@maxOccurs](#) [152], [xs:occurs/@maxOccurs](#) [365]
[xs:narrowMaxMin/@maxOccurs](#) [217],

Known Usage Locations

- As direct type of attributes within `attributeGroups` (1):
[xs:occurs/@maxOccurs](#) [365]
- In derivations of anonymous types of attributes within `complexType` (2):
[xs:narrowMaxMin/@maxOccurs](#) [217] (as restriction base)
[xs:all/@maxOccurs](#) [152] (as restriction base)

Annotation

for `maxOccurs`

Type Definition Detail

Type Derivation Tree

union of ([xs:nonNegativeInteger](#) | restriction of [xs:NMTOKEN](#))

└─ **xs:allNNI**

Derivation: by union

Member Types

1. [xs:nonNegativeInteger](#)
2. anonymous simpleType:
Derivation: restriction of [xs:NMTOKEN](#)
Facets: enumeration: "unbounded"

XML Source (see within schema source: p. 400)

```
<xs:simpleType name="allNNI">
  <xs:annotation>
    <xs:documentation>
      for maxOccurs
    </xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="xs:nonNegativeInteger">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="unbounded"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

simpleType "xs:allNNI"

</xs:simpleType>

simpleType

xs:anyURI

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [265]

Simple Content Model

[xs:anyURI](#)

Simple Content Restrictions:

WhiteSpace: collapse

Known Direct Subtypes (1):

[xs:namespaceList](#) [306]

All Direct / Indirect Based Attributes (10):

xml:base [368],	xs:include/@schemaLocation [84],
xs:appinfo/@source [26],	xs:notation/@system [107],
xs:documentation/@source [52],	xs:redefine/@schemaLocation [112],
xs:import/@namespace [82],	xs:schema/@targetNamespace [10],
xs:import/@schemaLocation [82],	xs:wildcard/@namespace [261]

Known Usage Locations

- In derivations of other global types (1):

[xs:namespaceList](#) [306] (as union member)

- As direct type of global attributes (1):

[xml:base](#) [368]

- As direct type of attributes within elements (8):

xs:appinfo/@source [26],	xs:include/@schemaLocation [84],
xs:documentation/@source [52],	xs:notation/@system [107],
xs:import/@namespace [82],	xs:redefine/@schemaLocation [112],
xs:import/@schemaLocation [82],	xs:schema/@targetNamespace [10]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#anyURI>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:anyURI
```

Derivation: restriction of [xs:anySimpleType](#)

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 418)

```
<xs:simpleType id="anyURI" name="anyURI">
  <xs:annotation>
    <xs:appinfo>
```

```
<hfp:hasFacet name="length"/>
<hfp:hasFacet name="minLength"/>
<hfp:hasFacet name="maxLength"/>
<hfp:hasFacet name="pattern"/>
<hfp:hasFacet name="enumeration"/>
<hfp:hasFacet name="whiteSpace"/>
<hfp:hasProperty name="ordered" value="false"/>
<hfp:hasProperty name="bounded" value="false"/>
<hfp:hasProperty name="cardinality" value="countably infinite"/>
<hfp:hasProperty name="numeric" value="false"/>
</xs:appinfo>
<xs:documentation source="http://www.w3.org/TR/xmlschema-2/#anyURI"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="anyURI.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
```

simpleType

xs:base64Binary

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [267]

Simple Content Model

[xs:base64Binary](#)

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#base64Binary>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)
└─ xs:base64Binary

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 418)

```
<xs:simpleType id="base64Binary" name="base64Binary">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#base64Binary"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="base64Binary.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:blockSet

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [268]

Simple Content Model

"#all" | list of ("extension" | "restriction" | "substitution")

All Direct / Indirect Based Attributes (2):

[xs:element/@block](#) [176], [xs:schema/@blockDefault](#) [9]

Known Usage Locations

- As direct type of attributes within elements (1):
[xs:schema/@blockDefault](#) [9]
- As direct type of attributes within complexTypes (1):
[xs:element/@block](#) [176]

Annotation

Annotation 1 [[src](#), p. 404]:

A utility type, not for public use

Annotation 2 [[src](#), p. 404]:

#all or (possibly empty) subset of {substitution, extension, restriction}

Type Definition Detail

Type Derivation Tree

union of (restriction of [xs:token](#) | list of restriction of [xs:derivationControl](#))
└─ [xs:blockSet](#)

Derivation: **by union**

Member Types

1. anonymous simpleType:
Derivation: **restriction of** [xs:token](#)
Facets: enumeration: "#all"
2. anonymous simpleType:
Derivation: **list of** anonymous simpleType
[Anonymous simpleType](#)
Derivation: **restriction of** [xs:derivationControl](#)
Facets: enumeration: "extension", "restriction", "substitution"

XML Source ([see](#) within schema source: p. 404)

```
<xs:simpleType name="blockSet">  
  <xs:annotation>  
    <xs:documentation>  
      A utility type, not for public use  
    </xs:documentation>  
  </xs:annotation>  
</xs:simpleType>
```



```
#all or (possibly empty) subset of {substitution, extension,  
restriction}  
</xs:documentation>  
</xs:annotation>  
<xs:union>  
  <xs:simpleType>  
    <xs:restriction base="xs:token">  
      <xs:enumeration value="#all"/>  
    </xs:restriction>  
  </xs:simpleType>  
<xs:simpleType>  
  <xs:list>  
    <xs:simpleType>  
      <xs:restriction base="xs:derivationControl">  
        <xs:enumeration value="extension"/>  
        <xs:enumeration value="restriction"/>  
        <xs:enumeration value="substitution"/>  
      </xs:restriction>  
    </xs:simpleType>  
  </xs:list>  
</xs:simpleType>  
</xs:union>  
</xs:simpleType>
```

simpleType xs:boolean

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [270]

Simple Content Model

xs:boolean

Simple Content Restrictions:

WhiteSpace: collapse

All Direct / Indirect Based Attributes (6):

[xs:complexContent/@mixed](#) [42], [xs:element/@abstract](#) [176],
[xs:complexType/@abstract](#) [172], [xs:element/@nillable](#) [177],
[xs:complexContent/@mixed](#) [172], [xs:facet/@fixed](#) [187]

Known Usage Locations

- As direct type of attributes within elements (1):

[xs:complexContent/@mixed](#) [42]

- As direct type of attributes within complexTypes (5):

[xs:complexType/@abstract](#) [172], [xs:element/@nillable](#) [177],
[xs:complexContent/@mixed](#) [172], [xs:facet/@fixed](#) [187]
[xs:element/@abstract](#) [176],

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#boolean>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:boolean
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 414)

```
<xs:simpleType id="boolean" name="boolean">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#boolean"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="boolean.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType "xs:boolean"

</xs:simpleType>

simpleType xs:byte

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [272]

Simple Content Model

[xs:byte](#)

Simple Content Restrictions:

MinInclusive: -128
MaxInclusive: 127

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#byte>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       ├── xs:long [303] (restriction)
│       │   └── xs:int [298] (restriction)
│       │       └── xs:short [325] (restriction)
│       │           └── xs:byte
```

Derivation: restriction of [xs:short](#)

Facets: minInclusive: -128
maxInclusive: 127

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="byte" name="byte">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#byte"/>
  </xs:annotation>
  <xs:restriction base="xs:short">
    <xs:minInclusive id="byte.minInclusive" value="-128"/>
    <xs:maxInclusive id="byte.maxInclusive" value="127"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:date

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [273]

Simple Content Model

xs:date

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#date>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:date
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 416)

```
<xs:simpleType id="date" name="date">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#date"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="date.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:dateTime

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [274]

Simple Content Model

xs:dateTime

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#dateTime>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:dateTime
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 415)

```
<xs:simpleType id="dateTime" name="dateTime">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#dateTime"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="dateTime.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:decimal

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [275]

Simple Content Model

xs:decimal

Simple Content Restrictions:

WhiteSpace: collapse

Known Direct Subtypes (1):

[xs:integer](#) [299]

Known Indirect Subtypes (13):

[xs:allNNI](#) [263], [xs:byte](#) [272], [xs:int](#) [298], [xs:long](#) [303], [xs:negativeInteger](#) [310], [xs:nonNegativeInteger](#) [314],
[xs:nonPositiveInteger](#) [316], [xs:positiveInteger](#) [320], [xs:short](#) [325], [xs:unsignedByte](#) [334], [xs:unsignedInt](#) [335],
[xs:unsignedLong](#) [336], [xs:unsignedShort](#) [337]

All Direct / Indirect Based Attributes (8):

xs:all/@maxOccurs [152],	xs:numFacet/@value [223],
xs:all/@minOccurs [152],	xs:occurs/@maxOccurs [365],
xs:narrowMaxMin/@maxOccurs [217],	xs:occurs/@minOccurs [366],
xs:narrowMaxMin/@minOccurs [218],	xs:totalDigits/@value [141]

Known Usage Locations

- In derivations of other global types (1):

[xs:integer](#) [299] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#decimal>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:decimal
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 414)

```
<xs:simpleType id="decimal" name="decimal">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="totalDigits"/>
      <hfp:hasFacet name="fractionDigits"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="maxInclusive"/>
    </xs:appinfo>
  </xs:annotation>
</xs:simpleType>
```

```
<hfp:hasFacet name="maxExclusive"/>
<hfp:hasFacet name="minInclusive"/>
<hfp:hasFacet name="minExclusive"/>
<hfp:hasProperty name="ordered" value="total"/>
<hfp:hasProperty name="bounded" value="false"/>
<hfp:hasProperty name="cardinality" value="countably infinite"/>
<hfp:hasProperty name="numeric" value="true"/>
</xs:appinfo>
<xs:documentation source="http://www.w3.org/TR/xmlschema-2/#decimal"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="decimal.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
```


simpleType xs:derivationControl

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [277]

Simple Content Model

enumeration of [xs:NMTOKEN](#)

Simple Content Restrictions:

Enumeration: "substitution", "extension", "restriction", "list", "union"

Known Direct Subtypes (4):

[xs:blockSet](#) [268], [xs:reducedDerivationControl](#) [324], [xs:simpleDerivationSet](#) [326], [xs:typeDerivationControl](#) [333]

Known Indirect Subtypes (2):

[xs:derivationSet](#) [279], [xs:fullDerivationSet](#) [287]

All Direct / Indirect Based Attributes (7):

[xs:complexType/@block](#) [172], [xs:schema/@blockDefault](#) [9],
[xs:complexType/@final](#) [172], [xs:schema/@finalDefault](#) [9],
[xs:element/@block](#) [176], [xs:simpleType/@final](#) [244]
[xs:element/@final](#) [177],

Known Usage Locations

- In derivations of other global types (4):

[xs:blockSet](#) [268] (as restriction base),
[xs:reducedDerivationControl](#) [324] (as restriction base),
[xs:simpleDerivationSet](#) [326] (as restriction base),
[xs:typeDerivationControl](#) [333] (as restriction base)

Annotation

A utility type, not for public use

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   ├── xs:normalizedString [317] (restriction)
│   │   ├── xs:token [331] (restriction)
│   │   │   ├── xs:NMTOKEN [311] (restriction)
│   │   │   └── xs:derivationControl
```

Derivation: restriction of [xs:NMTOKEN](#)

Facets: enumeration: "substitution", "extension", "restriction", "list", "union"

XML Source (see within schema source: p. 423)

```
<xs:simpleType name="derivationControl">
  <xs:annotation>
```

```
<xs:documentation>  
  A utility type, not for public use  
</xs:documentation>  
</xs:annotation>  
<xs:restriction base="xs:NMTOKEN">  
  <xs:enumeration value="substitution"/>  
  <xs:enumeration value="extension"/>  
  <xs:enumeration value="restriction"/>  
  <xs:enumeration value="list"/>  
  <xs:enumeration value="union"/>  
</xs:restriction>  
</xs:simpleType>
```

simpleType

xs:derivationSet

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [279]

Simple Content Model

"#all" | list of ("extension" | "restriction")

All Direct / Indirect Based Attributes (3):

[xs:complexType/@block](#) [172], [xs:element/@final](#) [177]
[xs:complexType/@final](#) [172],

Known Usage Locations

- As direct type of attributes within complexTypes (3):

[xs:complexType/@block](#) [172], [xs:element/@final](#) [177]
[xs:complexType/@final](#) [172],

Annotation

Annotation 1 [src, p. 398]:

A utility type, not for public use

Annotation 2 [src, p. 398]:

#all or (possibly empty) subset of {extension, restriction}

Type Definition Detail

Type Derivation Tree

union of (restriction of [xs:token](#) | list of [xs:reducedDerivationControl](#))

└ [xs:derivationSet](#)

Derivation: by union

Member Types

- anonymous simpleType:
Derivation: restriction of [xs:token](#)
Facets: enumeration: "#all"
- anonymous simpleType:
Derivation: list of [xs:reducedDerivationControl](#)

XML Source (see within schema source: p. 398)

```
<xs:simpleType name="derivationSet">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
    <xs:documentation>
      #all or (possibly empty) subset of {extension, restriction}
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="#all"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

```
</xs:simpleType>  
<xs:simpleType>  
  <xs:list itemType="xs:reducedDerivationControl"/>  
</xs:simpleType>  
</xs:union>  
</xs:simpleType>
```

simpleType

xs:double

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [281]

Simple Content Model

xs:double

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#double>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)
└─ xs:double

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 414)

```
<xs:simpleType id="double" name="double">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="total"/>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
      <hfp:hasProperty name="numeric" value="true"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#double"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="double.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:duration

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [282]

Simple Content Model

`xs:duration`

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#duration>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:duration
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 415)

```
<xs:simpleType id="duration" name="duration">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#duration"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="duration.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:ENTITIES

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [283]

Simple Content Model

xs:ENTITIES

Simple Content Restrictions:

MinLength: 1

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#ENTITIES>

Type Definition Detail

Type Derivation Tree

```

xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:Name [304] (restriction)
│               └── xs:NCName [308] (restriction)
│                   └── xs:ENTITY [284] (restriction of list)
│                       └── xs:ENTITIES

```

Derivation: restriction of anonymous simpleType

Facets: minLength: 1

[Anonymous simpleType](#)

Derivation: list of xs:ENTITY

XML Source (see within schema source: p. 420)

```

<xs:simpleType id="ENTITIES" name="ENTITIES">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#ENTITIES"/>
  </xs:annotation>
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:ENTITY"/>
    </xs:simpleType>
    <xs:minLength id="ENTITIES.minLength" value="1"/>
  </xs:restriction>
</xs:simpleType>

```

simpleType

xs:ENTITY

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [284]

Simple Content Model

xs:ENTITY

Known Direct Subtypes (1):

[xs:ENTITIES](#) [283]

Known Usage Locations

- In derivations of other global types (1):

[xs:ENTITIES](#) [283] (as list item type)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#ENTITY>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           ├── xs:Name [304] (restriction)
│           └── xs:NCName [308] (restriction)
│               └── xs:ENTITY
```

Derivation: restriction of [xs:NCName](#)

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="ENTITY" name="ENTITY">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#ENTITY"/>
  </xs:annotation>
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
```


simpleType

xs:float

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [285]

Simple Content Model

xs:float

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#float>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:float
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 414)

```
<xs:simpleType id="float" name="float">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="total"/>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
      <hfp:hasProperty name="numeric" value="true"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#float"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="float.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:formChoice

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [286]

Simple Content Model

enumeration of [xs:NMTOKEN](#)

Simple Content Restrictions:

Enumeration: "qualified", "unqualified"

All Direct / Indirect Based Attributes (4):

[xs:attribute/@form](#) [160], [xs:schema/@attributeFormDefault](#) [9],
[xs:element/@form](#) [177], [xs:schema/@elementFormDefault](#) [9]

Known Usage Locations

- As direct type of attributes within elements (2):
[xs:schema/@attributeFormDefault](#) [9], [xs:schema/@elementFormDefault](#) [9]
- As direct type of attributes within complexTypes (2):
[xs:attribute/@form](#) [160], [xs:element/@form](#) [177]

Annotation

A utility type, not for public use

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:NMTOKEN [311] (restriction)
│               └── xs:formChoice
```

Derivation: [restriction](#) of [xs:NMTOKEN](#)
Facets: enumeration: "qualified", "unqualified"

XML Source (see within schema source: p. 398)

```
<xs:simpleType name="formChoice">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="qualified"/>
    <xs:enumeration value="unqualified"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:fullDerivationSet

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [287]

Simple Content Model

"#all" | list of ("extension" | "restriction" | "list" | "union")

All Direct / Indirect Based Attributes (1):

[xs:schema/@finalDefault](#) [9]

Known Usage Locations

- As direct type of attributes within elements (1):

[xs:schema/@finalDefault](#) [9]

Annotation

Annotation 1 [src, p. 399]:

A utility type, not for public use

Annotation 2 [src, p. 399]:

#all or (possibly empty) subset of {extension, restriction, list, union}

Type Definition Detail

Type Derivation Tree

union of (restriction of [xs:token](#) | list of [xs:typeDerivationControl](#))

└ [xs:fullDerivationSet](#)

Derivation: by union

Member Types

- anonymous simpleType:
 - Derivation: [restriction of xs:token](#)
 - Facets: enumeration: "#all"
- anonymous simpleType:
 - Derivation: [list of xs:typeDerivationControl](#)

XML Source (see within schema source: p. 399)

```
<xs:simpleType name="fullDerivationSet">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
    <xs:documentation>
      #all or (possibly empty) subset of {extension, restriction, list, union}
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="#all"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

```
<xs:list itemType="xs:typeDerivationControl" />  
</xs:simpleType>  
</xs:union>  
</xs:simpleType>
```

simpleType xs:gDay

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [289]

Simple Content Model

xs:gDay

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#gDay>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)

└─ xs:gDay

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 417)

```
<xs:simpleType id="gDay" name="gDay">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gDay"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gDay.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:gMonth

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [290]

Simple Content Model

[xs:gMonth](#)

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#gMonth>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:gMonth
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 417)

```
<xs:simpleType id="gMonth" name="gMonth">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gMonth"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gMonth.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:gMonthDay

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [291]

Simple Content Model

[xs:gMonthDay](#)

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#gMonthDay>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)
└─ xs:gMonthDay

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 417)

```
<xs:simpleType id="gMonthDay" name="gMonthDay">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gMonthDay"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gMonthDay.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:gYear

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [292]

Simple Content Model

xs:gYear

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#gYear>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)
└─ xs:gYear

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 416)

```
<xs:simpleType id="gYear" name="gYear">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gYear"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gYear.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```


simpleType

xs:gYearMonth

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [293]

Simple Content Model

[xs:gYearMonth](#)

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#gYearMonth>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)

└─ xs:gYearMonth

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 416)

```
<xs:simpleType id="gYearMonth" name="gYearMonth">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gYearMonth"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gYearMonth.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:hexBinary

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [294]

Simple Content Model

[xs:hexBinary](#)

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#binary>

Type Definition Detail

Type Derivation Tree

xs:anySimpleType (restriction)
└─ xs:hexBinary

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 417)

```
<xs:simpleType id="hexBinary" name="hexBinary">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#binary"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="hexBinary.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:ID

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [295]

Simple Content Model

xs:ID

All Direct / Indirect Based Attributes (5):

[xml:id](#) [369], [xs:redefine/@id](#) [112],
[xs:annotated/@id](#) [155], [xs:schema/@id](#) [10]
[xs:annotation/@id](#) [19],

Known Usage Locations

- As direct type of global attributes (1):
[xml:id](#) [369]
- As direct type of attributes within elements (3):
[xs:annotation/@id](#) [19], [xs:schema/@id](#) [10]
[xs:redefine/@id](#) [112],
- As direct type of attributes within complexTypes (1):
[xs:annotated/@id](#) [155]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#ID>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├─ xs:string [328] (restriction)
│   └─ xs:normalizedString [317] (restriction)
│       └─ xs:token [331] (restriction)
│           └─ xs:Name [304] (restriction)
│               └─ xs:NCName [308] (restriction)
│                   └─ xs:ID
```

Derivation: restriction of [xs:NCName](#)

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="ID" name="ID">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#ID"/>
  </xs:annotation>
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
```

simpleType

xs:IDREF

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [296]

Simple Content Model

xs:IDREF

Known Direct Subtypes (1):

[xs:IDREFS](#) [297]

Known Usage Locations

- In derivations of other global types (1):

[xs:IDREFS](#) [297] (as list item type)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#IDREF>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:Name [304] (restriction)
│               └── xs:NCName [308] (restriction)
│                   └── xs:IDREF
```

Derivation: restriction of [xs:NCName](#)

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="IDREF" name="IDREF">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#IDREF"/>
  </xs:annotation>
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
```

simpleType

xs:IDREFS

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [297]

Simple Content Model

xs:IDREFS

Simple Content Restrictions:

MinLength: 1

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#IDREFS>

Type Definition Detail

Type Derivation Tree

```

xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:Name [304] (restriction)
│               └── xs:NCName [308] (restriction)
│                   └── xs:IDREF [296] (restriction of list)
│                       └── xs:IDREFS

```

Derivation: restriction of anonymous simpleType

Facets: minLength: 1

[Anonymous simpleType](#)

Derivation: list of xs:IDREF

XML Source (see within schema source: p. 419)

```

<xs:simpleType id="IDREFS" name="IDREFS">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#IDREFS"/>
  </xs:annotation>
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:IDREF"/>
    </xs:simpleType>
    <xs:minLength id="IDREFS.minLength" value="1"/>
  </xs:restriction>
</xs:simpleType>

```

simpleType

xs:int

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [298]

Simple Content Model

`xs:int`

Simple Content Restrictions:

MinInclusive: -2147483648
MaxInclusive: 2147483647

Known Direct Subtypes (1):

[xs:short](#) [325]

Known Indirect Subtypes (1):

[xs:byte](#) [272]

Known Usage Locations

- In derivations of other global types (1):

[xs:short](#) [325] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#int>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:long [303] (restriction)
│           └── xs:int
```

Derivation: [restriction of xs:long](#)

Facets: minInclusive: -2147483648
maxInclusive: 2147483647

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="int" name="int">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#int"/>
  </xs:annotation>
  <xs:restriction base="xs:long">
    <xs:minInclusive id="int.minInclusive" value="-2147483648"/>
    <xs:maxInclusive id="int.maxInclusive" value="2147483647"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:integer

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [299]

Simple Content Model

xs:integer

Simple Content Restrictions:

FractionDigits: 0
Pattern: `[\-+]?[0-9]+`

Known Direct Subtypes (3):

[xs:long](#) [303], [xs:nonNegativeInteger](#) [314], [xs:nonPositiveInteger](#) [316]

Known Indirect Subtypes (10):

[xs:allNNI](#) [263], [xs:byte](#) [272], [xs:int](#) [298], [xs:negativeInteger](#) [310], [xs:positiveInteger](#) [320], [xs:short](#) [325],
[xs:unsignedByte](#) [334], [xs:unsignedInt](#) [335], [xs:unsignedLong](#) [336], [xs:unsignedShort](#) [337]

All Direct / Indirect Based Attributes (8):

xs:all/@maxOccurs [152],	xs:numFacet/@value [223],
xs:all/@minOccurs [152],	xs:occurs/@maxOccurs [365],
xs:narrowMaxMin/@maxOccurs [217],	xs:occurs/@minOccurs [366],
xs:narrowMaxMin/@minOccurs [218],	xs:totalDigits/@value [141]

Known Usage Locations

- In derivations of other global types (3):

[xs:long](#) [303] (as restriction base), [xs:nonPositiveInteger](#) [316] (as restriction base)
[xs:nonNegativeInteger](#) [314] (as restriction base),

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#integer>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer
```

Derivation: restriction of [xs:decimal](#)
Facets: fractionDigits: 0
pattern: `[\-+]?[0-9]+`

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="integer" name="integer">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#integer"/>
  </xs:annotation>
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits fixed="true" id="integer.fractionDigits" value="0"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:pattern value="\[-+\]?[0-9]+"\>  
</xs:restriction>  
</xs:simpleType>
```


simpleType xs:language

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [301]

Simple Content Model

[xs:language](#)

Simple Content Restrictions:

Pattern: `[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*`
pattern specifies the content of section 2.12 of XML 1.0e2 and RFC 3066 (Revised version of RFC 1766).

See: <http://www.ietf.org/rfc/rfc3066.txt>

All Direct / Indirect Based Attributes (1):

[xml:lang](#) [370]

Known Usage Locations

- In derivations of anonymous types of global attributes (1):

[xml:lang](#) [370] (as union member)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#language>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:language
```

Derivation: restriction of [xs:token](#)

Facets: **pattern:** `[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*`
pattern specifies the content of section 2.12 of XML 1.0e2 and RFC 3066 (Revised version of RFC 1766).

See: <http://www.ietf.org/rfc/rfc3066.txt>

XML Source (see within schema source: p. 419)

```
<xs:simpleType id="language" name="language">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#language"/>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:pattern id="language.pattern" value="[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*">
      <xs:annotation>
        <xs:documentation source="http://www.ietf.org/rfc/rfc3066.txt">
          pattern specifies the content of section 2.12 of XML 1.0e2
          and RFC 3066 (Revised version of RFC 1766).
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
```

```
</xs:restriction>  
</xs:simpleType>
```

simpleType

xs:long

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [303]

Simple Content Model

[xs:long](#)

Simple Content Restrictions:

MinInclusive: -9223372036854775808
MaxInclusive: 9223372036854775807

Known Direct Subtypes (1):

[xs:int](#) [298]

Known Indirect Subtypes (2):

[xs:byte](#) [272], [xs:short](#) [325]

Known Usage Locations

- In derivations of other global types (1):

[xs:int](#) [298] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#long>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:long
```

Derivation: restriction of [xs:integer](#)
Facets: **minInclusive:** -9223372036854775808
maxInclusive: 9223372036854775807

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="long" name="long">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#long"/>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:minInclusive id="long.minInclusive" value="-9223372036854775808"/>
    <xs:maxInclusive id="long.maxInclusive" value="9223372036854775807"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:Name

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [305]

Simple Content Model

`xs:Name`

Simple Content Restrictions:

Pattern: `\i\c*`
pattern matches production 5 from the XML spec
See: <http://www.w3.org/TR/REC-xml#NT-Name>

Known Direct Subtypes (1):

[xs:NCName](#) [308]

Known Indirect Subtypes (5):

[xs:ENTITIES](#) [283], [xs:ENTITY](#) [284], [xs:ID](#) [295], [xs:IDREF](#) [296], [xs:IDREFS](#) [297]

All Direct / Indirect Based Attributes (17):

xml:id [369],	xs:notation/@name [107],
xml:space [372],	xs:redefine/@id [112],
xs:annotated/@id [155],	xs:schema/@id [10],
xs:annotation/@id [19],	xs:simpleType/@name [244],
xs:complexType/@name [172],	xs:topLevelAttribute/@name [247],
xs:defRef/@name [363],	xs:topLevelComplexType/@name [251],
xs:keybase/@name [196],	xs:topLevelElement/@name [255],
xs:namedAttributeGroup/@name [210],	xs:topLevelSimpleType/@name [258]
xs:namedGroup/@name [214],	

Known Usage Locations

- In derivations of other global types (1):

[xs:NCName](#) [308] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#Name>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:Name
```

Derivation: restriction of [xs:token](#)
Facets: pattern: `\i\c*`
pattern matches production 5 from the XML spec
See: <http://www.w3.org/TR/REC-xml#NT-Name>

XML Source (see within schema source: p. 420)

```
<xs:simpleType id="Name" name="Name">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#Name"/>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:pattern id="Name.pattern" value="\i\c*">
      <xs:annotation>
        <xs:documentation source="http://www.w3.org/TR/REC-xml#NT-Name">
          pattern matches production 5 from the XML spec
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:namespaceList

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [306]

Simple Content Model

("##any" | "##other") | list of ([xs:anyURI](#) | ("##targetNamespace" | "##local"))

All Direct / Indirect Based Attributes (1):

[xs:wildcard/@namespace](#) [261]

Known Usage Locations

- As direct type of attributes within complexTypes (1):

[xs:wildcard/@namespace](#) [261]

Annotation

A utility type, not for public use

Type Definition Detail

Type Derivation Tree

union of (restriction of [xs:token](#) | list of union of ([xs:anyURI](#) | restriction of [xs:token](#)))

└ [xs:namespaceList](#)

Derivation: **by union**

Member Types

- anonymous simpleType:
Derivation: **restriction of [xs:token](#)**
Facets: **enumeration:** "##any", "##other"

- anonymous simpleType:
Derivation: **list of anonymous simpleType**

Anonymous simpleType

Derivation: **by union**

Member Types

- [xs:anyURI](#)
- anonymous simpleType:
Derivation: **restriction of [xs:token](#)**
Facets: **enumeration:** "##targetNamespace", "##local"

XML Source (see within schema source: p. 409)

```
<xs:simpleType name="namespaceList">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="##any"/>
        <xs:enumeration value="##other"/>
      </xs:restriction>
```

```
</xs:simpleType>
<xs:simpleType>
  <xs:list>
    <xs:simpleType>
      <xs:union memberTypes="xs:anyURI">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="##targetNamespace"/>
            <xs:enumeration value="##local"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
</xs:union>
</xs:simpleType>
```

simpleType xs:NCName

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [309]

Simple Content Model

[xs:NCName](#)

Simple Content Restrictions:

Pattern: `[\i-[:]][\c-[:]]*`

pattern matches production 4 from the Namespaces in XML spec

See: <http://www.w3.org/TR/REC-xml-names/#NT-NCName>

Known Direct Subtypes (3):

[xs:ENTITY](#) [284], [xs:ID](#) [295], [xs:IDREF](#) [296]

Known Indirect Subtypes (2):

[xs:ENTITIES](#) [283], [xs:IDREFS](#) [297]

All Direct / Indirect Based Attributes (17):

xml:id [369],	xs:notation/@name [107],
xml:space [372],	xs:redefine/@id [112],
xs:annotated/@id [155],	xs:schema/@id [10],
xs:annotation/@id [19],	xs:simpleType/@name [244],
xs:complexType/@name [172],	xs:topLevelAttribute/@name [247],
xs:defRef/@name [363],	xs:topLevelComplexType/@name [251],
xs:keybase/@name [196],	xs:topLevelElement/@name [255],
xs:namedAttributeGroup/@name [210],	xs:topLevelSimpleType/@name [258]
xs:namedGroup/@name [214],	

Known Usage Locations

- In derivations of other global types (3):

[xs:ENTITY](#) [284] (as restriction base), [xs:IDREF](#) [296] (as restriction base)
[xs:ID](#) [295] (as restriction base),

- As direct type of attributes within elements (1):

[xs:notation/@name](#) [107]

- As direct type of attributes within complexTypes (9):

xs:complexType/@name [172],	xs:topLevelAttribute/@name [247],
xs:keybase/@name [196],	xs:topLevelComplexType/@name [251],
xs:namedAttributeGroup/@name [210],	xs:topLevelElement/@name [255],
xs:namedGroup/@name [214],	xs:topLevelSimpleType/@name [258]
xs:simpleType/@name [244],	

- As direct type of attributes within attributeGroups (1):

[xs:defRef/@name](#) [363]

- In derivations of anonymous types of global attributes (1):

[xml:space](#) [372] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#NCName>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:Name [304] (restriction)
│               └── xs:NCName
```

Derivation: restriction of xs:Name

Facets: pattern: [\i-[:]][\c-[:]]*
pattern matches production 4 from the Namespaces in XML spec

See: <http://www.w3.org/TR/REC-xml-names/#NT-NCName>

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="NCName" name="NCName">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NCName"/>
  </xs:annotation>
  <xs:restriction base="xs:Name">
    <xs:pattern id="NCName.pattern" value="[\i-[:]][\c-[:]]*">
      <xs:annotation>
        <xs:documentation source="http://www.w3.org/TR/REC-xml-names/#NT-NCName">
          pattern matches production 4 from the Namespaces in XML spec
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:negativeInteger

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [310]

Simple Content Model

[xs:negativeInteger](#)

Simple Content Restrictions:

MaxInclusive: -1

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#negativeInteger>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:nonPositiveInteger [316] (restriction)
│           └── xs:negativeInteger
```

Derivation: restriction of [xs:nonPositiveInteger](#)

Facets: maxInclusive: -1

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="negativeInteger" name="negativeInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#negativeInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:nonPositiveInteger">
    <xs:maxInclusive id="negativeInteger.maxInclusive" value="-1"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:NMTOKEN

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [312]

Simple Content Model

xs:NMTOKEN

Simple Content Restrictions:

Pattern: \c+

pattern matches production 7 from the XML spec

See: <http://www.w3.org/TR/REC-xml#NT-Nmtoken>

Known Direct Subtypes (4):

[xs:NMTOKENS](#) [313], [xs:allNNI](#) [263], [xs:derivationControl](#) [277], [xs:formChoice](#) [286]

Known Indirect Subtypes (6):

[xs:blockSet](#) [268], [xs:derivationSet](#) [279], [xs:fullDerivationSet](#) [287], [xs:reducedDerivationControl](#) [324],
[xs:simpleDerivationSet](#) [326], [xs:typeDerivationControl](#) [333]

All Direct / Indirect Based Attributes (17):

xs:all/@maxOccurs [152],	xs:occurs/@maxOccurs [365],
xs:attribute/@form [160],	xs:schema/@attributeFormDefault [9],
xs:attribute/@use [160],	xs:schema/@blockDefault [9],
xs:complexType/@block [172],	xs:schema/@elementFormDefault [9],
xs:complexType/@final [172],	xs:schema/@finalDefault [9],
xs:element/@block [176],	xs:simpleType/@final [244],
xs:element/@final [177],	xs:whiteSpace/@value [148],
xs:element/@form [177],	xs:wildcard/@processContents [261]
xs:narrowMaxMin/@maxOccurs [217],	

Known Usage Locations

- In derivations of other global types (4):

[xs:NMTOKENS](#) [313] (as list item type), [xs:derivationControl](#) [277] (as restriction base),
[xs:allNNI](#) [263] (as restriction base), [xs:formChoice](#) [286] (as restriction base)

- In derivations of anonymous types of attributes within elements (1):

[xs:whiteSpace/@value](#) [148] (as restriction base)

- In derivations of anonymous types of attributes within complexTypes (2):

[xs:attribute/@use](#) [160] (as restriction base)
[xs:wildcard/@processContents](#) [261] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#NMTOKEN>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:NMTOKEN
```

Derivation: restriction of xs:token

Facets: pattern: \c+
pattern matches production 7 from the XML spec
See: <http://www.w3.org/TR/REC-xml#NT-Nmtoken>

XML Source (see within schema source: p. 420)

```
<xs:simpleType id="NMTOKEN" name="NMTOKEN">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NMTOKEN"/>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:pattern id="NMTOKEN.pattern" value="\c+">
      <xs:annotation>
        <xs:documentation source="http://www.w3.org/TR/REC-xml#NT-Nmtoken">
          pattern matches production 7 from the XML spec
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:NMTOKENS

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [313]

Simple Content Model

xs:NMTOKENS

Simple Content Restrictions:

MinLength: 1

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#NMTOKENS>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   ├── xs:normalizedString [317] (restriction)
│   │   ├── xs:token [331] (restriction)
│   │   │   ├── xs:NMTOKEN [311] (restriction of list)
│   │   │   └── xs:NMTOKENS
```

Derivation: restriction of anonymous simpleType

Facets: minLength: 1

[Anonymous simpleType](#)

Derivation: list of xs:NMTOKEN

XML Source (see within schema source: p. 420)

```
<xs:simpleType id="NMTOKENS" name="NMTOKENS">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NMTOKENS"/>
  </xs:annotation>
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:NMTOKEN"/>
    </xs:simpleType>
    <xs:minLength id="NMTOKENS.minLength" value="1"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:nonNegativeInteger

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [315]

Simple Content Model

[xs:nonNegativeInteger](#)

Simple Content Restrictions:

MinInclusive: 0

Known Direct Subtypes (3):

[xs:allNNI](#) [263], [xs:positiveInteger](#) [320], [xs:unsignedLong](#) [336]

Known Indirect Subtypes (3):

[xs:unsignedByte](#) [334], [xs:unsignedInt](#) [335], [xs:unsignedShort](#) [337]

All Direct / Indirect Based Attributes (8):

[xs:all/@maxOccurs](#) [152], [xs:numFacet/@value](#) [223],
[xs:all/@minOccurs](#) [152], [xs:occurs/@maxOccurs](#) [365],
[xs:narrowMaxMin/@maxOccurs](#) [217], [xs:occurs/@minOccurs](#) [366],
[xs:narrowMaxMin/@minOccurs](#) [218], [xs:totalDigits/@value](#) [141]

Known Usage Locations

- In derivations of other global types (3):

[xs:allNNI](#) [263] (as union member), [xs:unsignedLong](#) [336] (as restriction base),
[xs:positiveInteger](#) [320] (as restriction base),

- As direct type of attributes within complexTypes (1):

[xs:numFacet/@value](#) [223]

- As direct type of attributes within attributeGroups (1):

[xs:occurs/@minOccurs](#) [366]

- In derivations of anonymous types of attributes within complexTypes (2):

[xs:narrowMaxMin/@minOccurs](#) [218] (as restriction base),
[xs:all/@minOccurs](#) [152] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#nonNegativeInteger>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:nonNegativeInteger
```

Derivation: restriction of [xs:integer](#)

Facets: minInclusive: 0

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="nonNegativeInteger" name="nonNegativeInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#nonNegativeInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:minInclusive id="nonNegativeInteger.minInclusive" value="0"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:nonPositiveInteger

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [316]

Simple Content Model

[xs:nonPositiveInteger](#)

Simple Content Restrictions:

MaxInclusive: 0

Known Direct Subtypes (1):

[xs:negativeInteger](#) [310]

Known Usage Locations

- In derivations of other global types (1):

[xs:negativeInteger](#) [310] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#nonPositiveInteger>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:nonPositiveInteger
```

Derivation: restriction of [xs:integer](#)

Facets: maxInclusive: 0

XML Source (see within schema source: p. 421)

```
<xs:simpleType id="nonPositiveInteger" name="nonPositiveInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#nonPositiveInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:maxInclusive id="nonPositiveInteger.maxInclusive" value="0"/>
  </xs:restriction>
</xs:simpleType>
```


simpleType

xs:normalizedString

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [318]

Simple Content Model

[xs:normalizedString](#)

Simple Content Restrictions:

WhiteSpace: `replace`

Known Direct Subtypes (1):

[xs:token](#) [331]

Known Indirect Subtypes (21):

[xs:ENTITIES](#) [283], [xs:ENTITY](#) [284], [xs:ID](#) [295], [xs:IDREF](#) [296], [xs:IDREFS](#) [297], [xs:NCName](#) [308], [xs:NMTOKEN](#) [311], [xs:NMTOKENS](#) [313], [xs:Name](#) [304], [xs:allNNI](#) [263], [xs:blockSet](#) [268], [xs:derivationControl](#) [277], [xs:derivationSet](#) [279], [xs:formChoice](#) [286], [xs:fullDerivationSet](#) [287], [xs:language](#) [301], [xs:namespaceList](#) [306], [xs:public](#) [321], [xs:reducedDerivationControl](#) [324], [xs:simpleDerivationSet](#) [326], [xs:typeDerivationControl](#) [333]

All Direct / Indirect Based Attributes (40):

xml:id [369],	xs:notation/@name [107],
xml:lang [370],	xs:notation/@public [107],
xml:space [372],	xs:occurs/@maxOccurs [365],
xs:all/@maxOccurs [152],	xs:redefine/@id [112],
xs:annotated/@id [155],	xs:schema/@attributeFormDefault [9],
xs:annotation/@id [19],	xs:schema/@blockDefault [9],
xs:attribute/@form [160],	xs:schema/@elementFormDefault [9],
xs:attribute/@use [160],	xs:schema/@finalDefault [9],
xs:complexType/@block [172],	xs:schema/@id [10],
xs:complexType/@final [172],	xs:schema/@version [10],
xs:complexType/@name [172],	xs:selector/@xpath [126],
xs:defRef/@name [363],	xs:simpleType/@final [244],
xs:element/@block [176],	xs:simpleType/@name [244],
xs:element/@final [177],	xs:topLevelAttribute/@name [247],
xs:element/@form [177],	xs:topLevelComplexType/@name [251],
xs:field/@xpath [73],	xs:topLevelElement/@name [255],
xs:keybase/@name [196],	xs:topLevelSimpleType/@name [258],
xs:namedAttributeGroup/@name [210],	xs:whiteSpace/@value [148],
xs:namedGroup/@name [214],	xs:wildcard/@namespace [261],
xs:narrowMaxMin/@maxOccurs [217],	xs:wildcard/@processContents [261]

Known Usage Locations

- In derivations of other global types (1):

[xs:token](#) [331] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#normalizedString>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString
```

Derivation: restriction of xs:string

Facets: whiteSpace: replace

XML Source (see within schema source: p. 419)

```
<xs:simpleType id="normalizedString" name="normalizedString">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#normalizedString"/>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace id="normalizedString.whiteSpace" value="replace"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:NOTATION

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [319]

Simple Content Model

xs:NOTATION

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

Annotation 1 [[src](#), p. 419]:

See: <http://www.w3.org/TR/xmlschema-2/#NOTATION>

Annotation 2 [[src](#), p. 419]:

NOTATION cannot be used directly in a schema; rather a type must be derived from it by specifying at least one enumeration facet whose value is the name of a NOTATION declared in the schema.

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:NOTATION
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 418)

```
<xs:simpleType id="NOTATION" name="NOTATION">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NOTATION"/>
    <xs:documentation>
      NOTATION cannot be used directly in a schema; rather a type
      must be derived from it by specifying at least one enumeration
      facet whose value is the name of a NOTATION declared in the
      schema.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="NOTATION.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:positiveInteger

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [320]

Simple Content Model

[xs:positiveInteger](#)

Simple Content Restrictions:

MinInclusive: 1

All Direct / Indirect Based Attributes (1):

[xs:totalDigits/@value](#) [141]

Known Usage Locations

- As direct type of attributes within elements (1):

[xs:totalDigits/@value](#) [141]

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#positiveInteger>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:nonNegativeInteger [314] (restriction)
│           └── xs:positiveInteger
```

Derivation: restriction of [xs:nonNegativeInteger](#)

Facets: minInclusive: 1

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="positiveInteger" name="positiveInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#positiveInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minInclusive id="positiveInteger.minInclusive" value="1"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType xs:public

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [321]

Simple Content Model

[xs:token](#)

All Direct / Indirect Based Attributes (1):

[xs:notation/@public](#) [107]

Known Usage Locations

- As direct type of attributes within elements (1):

[xs:notation/@public](#) [107]

Annotation

Annotation 1 [[src](#), p. 412]:

A utility type, not for public use

Annotation 2 [[src](#), p. 412]:

A public identifier, per ISO 8879

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:public
```

Derivation: restriction of [xs:token](#)

XML Source ([see within schema source](#): p. 412)

```
<xs:simpleType name="public">
  <xs:annotation>
    <xs:documentation>
      <b>A utility type, not for public use</b>
    </xs:documentation>
    <xs:documentation>
      <b>A public identifier, per ISO 8879</b>
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token" />
</xs:simpleType>
```

simpleType xs:QName

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [322]

Simple Content Model

[xs:QName](#)

Simple Content Restrictions:

WhiteSpace: collapse

All Direct / Indirect Based Attributes (12):

xs:attribute/@type [160],	xs:groupRef/@ref [194],
xs:attributeGroupRef/@ref [166],	xs:keyref/@refer [88],
xs:defRef/@ref [364],	xs:list/@itemType [93],
xs:element/@substitutionGroup [178],	xs:restriction/@base [116],
xs:element/@type [178],	xs:restrictionType/@base [231],
xs:extensionType/@base [184],	xs:union/@memberTypes [143]

Known Usage Locations

- As direct type of attributes within elements (3):

[xs:keyref/@refer](#) [88], [xs:restriction/@base](#) [116]
[xs:list/@itemType](#) [93],

- As direct type of attributes within complexTypes (7):

[xs:attribute/@type](#) [160], [xs:extensionType/@base](#) [184],
[xs:attributeGroupRef/@ref](#) [166], [xs:groupRef/@ref](#) [194],
[xs:element/@substitutionGroup](#) [178], [xs:restrictionType/@base](#) [231]
[xs:element/@type](#) [178],

- As direct type of attributes within attributeGroups (1):

[xs:defRef/@ref](#) [364]

- In derivations of anonymous types of attributes within elements (1):

[xs:union/@memberTypes](#) [143] (as list item type)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#QName>

Type Definition Detail

Type Derivation Tree

[xs:anySimpleType](#) (restriction)
└─ **xs:QName**

Derivation: restriction of [xs:anySimpleType](#)

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 418)

```
<xs:simpleType id="QName" name="QName">
```

```
<xs:annotation>
  <xs:appinfo>
    <hfp:hasFacet name="length"/>
    <hfp:hasFacet name="minLength"/>
    <hfp:hasFacet name="maxLength"/>
    <hfp:hasFacet name="pattern"/>
    <hfp:hasFacet name="enumeration"/>
    <hfp:hasFacet name="whiteSpace"/>
    <hfp:hasProperty name="ordered" value="false"/>
    <hfp:hasProperty name="bounded" value="false"/>
    <hfp:hasProperty name="cardinality" value="countably infinite"/>
    <hfp:hasProperty name="numeric" value="false"/>
  </xs:appinfo>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#QName"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="QName.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
```

simpleType xs:reducedDerivationControl

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [324]

Simple Content Model

enumeration of [xs:NMTOKEN](#)

Simple Content Restrictions:

Enumeration: "extension", "restriction"

Known Direct Subtypes (1):

[xs:derivationSet](#) [279]

All Direct / Indirect Based Attributes (3):

[xs:complexType/@block](#) [172], [xs:element/@final](#) [177]
[xs:complexType/@final](#) [172],

Known Usage Locations

- In derivations of other global types (1):

[xs:derivationSet](#) [279] (as list item type)

Annotation

A utility type, not for public use

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:NMTOKEN [311] (restriction)
│               └── xs:derivationControl [277] (restriction)
│                   └── xs:reducedDerivationControl
```

Derivation: restriction of [xs:derivationControl](#)
Facets: enumeration: "extension", "restriction"

XML Source (see within schema source: p. 398)

```
<xs:simpleType name="reducedDerivationControl">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:derivationControl">
    <xs:enumeration value="extension"/>
    <xs:enumeration value="restriction"/>
  </xs:restriction>
</xs:simpleType>
```


simpleType

xs:short

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [325]

Simple Content Model

[xs:short](#)

Simple Content Restrictions:

MinInclusive: -32768
MaxInclusive: 32767

Known Direct Subtypes (1):

[xs:byte](#) [272]

Known Usage Locations

- In derivations of other global types (1):
[xs:byte](#) [272] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#short>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       ├── xs:long [303] (restriction)
│       └── xs:int [298] (restriction)
│           └── xs:short
```

Derivation: restriction of [xs:int](#)
Facets: minInclusive: -32768
maxInclusive: 32767

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="short" name="short">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#short"/>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:minInclusive id="short.minInclusive" value="-32768"/>
    <xs:maxInclusive id="short.maxInclusive" value="32767"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:simpleDerivationSet

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [326]

Simple Content Model

"#all" | list of ("list" | "union" | "restriction")

All Direct / Indirect Based Attributes (1):

[xs:simpleType/@final](#) [244]

Known Usage Locations

- As direct type of attributes within complexTypes (1):

[xs:simpleType/@final](#) [244]

Annotation

Annotation 1 [src, p. 423]:

#all or (possibly empty) subset of {restriction, union, list}

Annotation 2 [src, p. 423]:

A utility type, not for public use

Type Definition Detail

Type Derivation Tree

union of (restriction of [xs:token](#) | list of restriction of [xs:derivationControl](#))

└ [xs:simpleDerivationSet](#)

Derivation: by union

Member Types

- anonymous simpleType:
Derivation: restriction of [xs:token](#)
Facets: enumeration: "#all"
- anonymous simpleType:
Derivation: list of anonymous simpleType
[Anonymous simpleType](#)
Derivation: restriction of [xs:derivationControl](#)
Facets: enumeration: "list", "union", "restriction"

XML Source (see within schema source: p. 423)

```
<xs:simpleType name="simpleDerivationSet">
  <xs:annotation>
    <xs:documentation>
      #all or (possibly empty) subset of {restriction, union, list}
    </xs:documentation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
```

```
<xs:enumeration value="#all"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:derivationControl">
        <xs:enumeration value="list"/>
        <xs:enumeration value="union"/>
        <xs:enumeration value="restriction"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
</xs:union>
</xs:simpleType>
```

simpleType

xs:string

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [329]

Simple Content Model

[xs:string](#)

Simple Content Restrictions:

WhiteSpace: `preserve`

Known Direct Subtypes (1):

[xs:normalizedString](#) [317]

Known Indirect Subtypes (22):

[xs:ENTITIES](#) [283], [xs:ENTITY](#) [284], [xs:ID](#) [295], [xs:IDREF](#) [296], [xs:IDREFS](#) [297], [xs:NCName](#) [308], [xs:NMTOKEN](#) [311], [xs:NMTOKENS](#) [313], [xs:Name](#) [304], [xs:allNNI](#) [263], [xs:blockSet](#) [268], [xs:derivationControl](#) [277], [xs:derivationSet](#) [279], [xs:formChoice](#) [286], [xs:fullDerivationSet](#) [287], [xs:language](#) [301], [xs:namespaceList](#) [306], [xs:public](#) [321], [xs:reducedDerivationControl](#) [324], [xs:simpleDerivationSet](#) [326], [xs:token](#) [331], [xs:typeDerivationControl](#) [333]

All Direct / Indirect Based Attributes (45):

xml:id [369],	xs:narrowMaxMin/@maxOccurs [217],
xml:lang [370],	xs:notation/@name [107],
xml:space [372],	xs:notation/@public [107],
xs:all/@maxOccurs [152],	xs:occurs/@maxOccurs [365],
xs:annotated/@id [155],	xs:pattern/@value [110],
xs:annotation/@id [19],	xs:redefine/@id [112],
xs:attribute/@default [160],	xs:schema/@attributeFormDefault [9],
xs:attribute/@fixed [160],	xs:schema/@blockDefault [9],
xs:attribute/@form [160],	xs:schema/@elementFormDefault [9],
xs:attribute/@use [160],	xs:schema/@finalDefault [9],
xs:complexType/@block [172],	xs:schema/@id [10],
xs:complexType/@final [172],	xs:schema/@version [10],
xs:complexType/@name [172],	xs:selector/@xpath [126],
xs:defRef/@name [363],	xs:simpleType/@final [244],
xs:element/@block [176],	xs:simpleType/@name [244],
xs:element/@default [177],	xs:topLevelAttribute/@name [247],
xs:element/@final [177],	xs:topLevelComplexType/@name [251],
xs:element/@fixed [177],	xs:topLevelElement/@name [255],
xs:element/@form [177],	xs:topLevelSimpleType/@name [258],
xs:field/@xpath [73],	xs:whiteSpace/@value [148],
xs:keybase/@name [196],	xs:wildcard/@namespace [261],
xs:namedAttributeGroup/@name [210],	xs:wildcard/@processContents [261]
xs:namedGroup/@name [214],	

Known Usage Locations

- In derivations of other global types (1):
 - [xs:normalizedString](#) [317] (as restriction base)
- As direct type of attributes within elements (1):
 - [xs:pattern/@value](#) [110]

- **As direct type of attributes within complexTypes (4):**
[xs:attribute/@default](#) [160], [xs:element/@default](#) [177],
[xs:attribute/@fixed](#) [160], [xs:element/@fixed](#) [177]
- **In derivations of anonymous types of global attributes (1):**
[xml:lang](#) [370] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#string>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:string
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: preserve

XML Source (see within schema source: p. 413)

```
<xs:simpleType id="string" name="string">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#string"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace id="string.preserve" value="preserve"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:time

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [330]

Simple Content Model

xs:time

Simple Content Restrictions:

WhiteSpace: collapse

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#time>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
└─ xs:time
```

Derivation: restriction of xs:anySimpleType

Facets: whiteSpace: collapse

XML Source (see within schema source: p. 415)

```
<xs:simpleType id="time" name="time">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#time"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="time.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:token

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [332]

Simple Content Model

`xs:token`

Simple Content Restrictions:

WhiteSpace: collapse

Known Direct Subtypes (9):

[xs:NMTOKEN](#) [311], [xs>Name](#) [304], [xs:blockSet](#) [268], [xs:derivationSet](#) [279], [xs:fullDerivationSet](#) [287],
[xs:language](#) [301], [xs:namespaceList](#) [306], [xs:public](#) [321], [xs:simpleDerivationSet](#) [326]

Known Indirect Subtypes (12):

[xs:ENTITIES](#) [283], [xs:ENTITY](#) [284], [xs:ID](#) [295], [xs:IDREF](#) [296], [xs:IDREFS](#) [297], [xs:NCName](#) [308],
[xs:NMTOKENS](#) [313], [xs:allNNI](#) [263], [xs:derivationControl](#) [277], [xs:formChoice](#) [286],
[xs:reducedDerivationControl](#) [324], [xs:typeDerivationControl](#) [333]

All Direct / Indirect Based Attributes (40):

xml:id [369],	xs:notation/@name [107],
xml:lang [370],	xs:notation/@public [107],
xml:space [372],	xs:occurs/@maxOccurs [365],
xs:all/@maxOccurs [152],	xs:redefine/@id [112],
xs:annotated/@id [155],	xs:schema/@attributeFormDefault [9],
xs:annotation/@id [19],	xs:schema/@blockDefault [9],
xs:attribute/@form [160],	xs:schema/@elementFormDefault [9],
xs:attribute/@use [160],	xs:schema/@finalDefault [9],
xs:complexType/@block [172],	xs:schema/@id [10],
xs:complexType/@final [172],	xs:schema/@version [10],
xs:complexType/@name [172],	xs:selector/@xpath [126],
xs:defRef/@name [363],	xs:simpleType/@final [244],
xs:element/@block [176],	xs:simpleType/@name [244],
xs:element/@final [177],	xs:topLevelAttribute/@name [247],
xs:element/@form [177],	xs:topLevelComplexType/@name [251],
xs:field/@xpath [73],	xs:topLevelElement/@name [255],
xs:keybase/@name [196],	xs:topLevelSimpleType/@name [258],
xs:namedAttributeGroup/@name [210],	xs:whiteSpace/@value [148],
xs:namedGroup/@name [214],	xs:wildcard/@namespace [261],
xs:narrowMaxMin/@maxOccurs [217],	xs:wildcard/@processContents [261]

Known Usage Locations

- In derivations of other global types (10):

xs:NMTOKEN [311] (as restriction base),	xs:language [301] (as restriction base),
xs>Name [304] (as restriction base),	xs:namespaceList [306] (as restriction base),
xs:blockSet [268] (as restriction base),	xs:namespaceList [306] (as restriction base),
xs:derivationSet [279] (as restriction base),	xs:public [321] (as restriction base),
xs:fullDerivationSet [287] (as restriction base),	xs:simpleDerivationSet [326] (as restriction base)

- As direct type of attributes within elements (1):

[xs:schema/@version](#) [10]

- In derivations of anonymous types of attributes within elements (2):

`xs:selector/@xpath` [126] (as restriction base)

`xs:field/@xpath` [73] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#token>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token
```

Derivation: restriction of `xs:normalizedString`

Facets: `whiteSpace`: collapse

XML Source (see within schema source: p. 419)

```
<xs:simpleType id="token" name="token">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#token"/>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:whiteSpace id="token.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
```


simpleType xs:typeDerivationControl

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [333]

Simple Content Model

enumeration of [xs:NMTOKEN](#)

Simple Content Restrictions:

Enumeration: "extension", "restriction", "list", "union"

Known Direct Subtypes (1):

[xs:fullDerivationSet](#) [287]

All Direct / Indirect Based Attributes (1):

[xs:schema/@finalDefault](#) [9]

Known Usage Locations

- In derivations of other global types (1):

[xs:fullDerivationSet](#) [287] (as list item type)

Annotation

A utility type, not for public use

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   ├── xs:normalizedString [317] (restriction)
│   │   ├── xs:token [331] (restriction)
│   │   │   ├── xs:NMTOKEN [311] (restriction)
│   │   │   └── xs:derivationControl [277] (restriction)
│   │       └── xs:typeDerivationControl
```

Derivation: [restriction](#) of [xs:derivationControl](#)

Facets: enumeration: "extension", "restriction", "list", "union"

XML Source (see within schema source: p. 398)

```
<xs:simpleType name="typeDerivationControl">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:derivationControl">
    <xs:enumeration value="extension"/>
    <xs:enumeration value="restriction"/>
    <xs:enumeration value="list"/>
    <xs:enumeration value="union"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:unsignedByte

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [334]

Simple Content Model

[xs:unsignedByte](#)

Simple Content Restrictions:

MaxInclusive: 255

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#unsignedByte>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       ├── xs:nonNegativeInteger [314] (restriction)
│       │   └── xs:unsignedLong [336] (restriction)
│       │       └── xs:unsignedInt [335] (restriction)
│       │           └── xs:unsignedShort [337] (restriction)
│       │               └── xs:unsignedByte
```

Derivation: restriction of [xs:unsignedShort](#)

Facets: maxInclusive: 255

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="unsignedByte" name="unsignedByte">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedByte"/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedShort">
    <xs:maxInclusive id="unsignedByte.maxInclusive" value="255"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:unsignedInt

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [335]

Simple Content Model

[xs:unsignedInt](#)

Simple Content Restrictions:

MaxInclusive: 4294967295

Known Direct Subtypes (1):

[xs:unsignedShort](#) [337]

Known Indirect Subtypes (1):

[xs:unsignedByte](#) [334]

Known Usage Locations

- In derivations of other global types (1):

[xs:unsignedShort](#) [337] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#unsignedInt>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       ├── xs:nonNegativeInteger [314] (restriction)
│       │   └── xs:unsignedLong [336] (restriction)
│       │       └── xs:unsignedInt
```

Derivation: restriction of [xs:unsignedLong](#)

Facets: maxInclusive: 4294967295

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="unsignedInt" name="unsignedInt">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedInt"/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedLong">
    <xs:maxInclusive id="unsignedInt.maxInclusive" value="4294967295"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:unsignedLong

Namespace: <http://www.w3.org/2001/XMLSchema>
Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [336]

Simple Content Model

[xs:unsignedLong](#)

Simple Content Restrictions:

MaxInclusive: 18446744073709551615

Known Direct Subtypes (1):

[xs:unsignedInt](#) [335]

Known Indirect Subtypes (2):

[xs:unsignedByte](#) [334], [xs:unsignedShort](#) [337]

Known Usage Locations

- In derivations of other global types (1):

[xs:unsignedInt](#) [335] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#unsignedLong>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       └── xs:nonNegativeInteger [314] (restriction)
│           └── xs:unsignedLong
```

Derivation: restriction of [xs:nonNegativeInteger](#)

Facets: maxInclusive: 18446744073709551615

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="unsignedLong" name="unsignedLong">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedLong"/>
  </xs:annotation>
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:maxInclusive id="unsignedLong.maxInclusive" value="18446744073709551615"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType

xs:unsignedShort

Namespace: <http://www.w3.org/2001/XMLSchema>

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [337]

Simple Content Model

[xs:unsignedShort](#)

Simple Content Restrictions:

MaxInclusive: 65535

Known Direct Subtypes (1):

[xs:unsignedByte](#) [334]

Known Usage Locations

- In derivations of other global types (1):

[xs:unsignedByte](#) [334] (as restriction base)

Annotation

See: <http://www.w3.org/TR/xmlschema-2/#unsignedShort>

Type Definition Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:decimal [275] (restriction)
│   └── xs:integer [299] (restriction)
│       ├── xs:nonNegativeInteger [314] (restriction)
│       │   ├── xs:unsignedLong [336] (restriction)
│       │   └── xs:unsignedInt [335] (restriction)
│       └── xs:unsignedShort
```

Derivation: restriction of [xs:unsignedInt](#)

Facets: maxInclusive: 65535

XML Source (see within schema source: p. 422)

```
<xs:simpleType id="unsignedShort" name="unsignedShort">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedShort"/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedInt">
    <xs:maxInclusive id="unsignedShort.maxInclusive" value="65535"/>
  </xs:restriction>
</xs:simpleType>
```

Element Groups

group

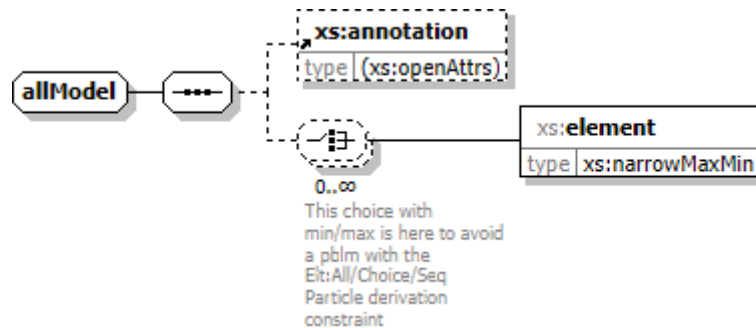
xs:allModel

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 2 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [338]

Component Diagram



Complex Content Model

[xs:annotation?](#), [xs:element*](#)

Content Model Elements (2):

[xs:annotation](#) [17], [xs:element](#) (type [xs:narrowMaxMin](#)) [61]

Known Usage Locations

- In definitions of global complexTypes (1):
[xs:all](#) [150]
- In definitions of anonymous complexTypes of elements (1):
[xs:all](#) (in [xs:group](#)) [15]

XML Source (see within schema source: p. 407)

```
<xs:group name="allModel">
  <xs:sequence>
    <xs:element minOccurs="0" ref="xs:annotation"/>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          This choice with min/max is here to
          avoid a pblm with the EIt:All/Choice/Seq
          Particle derivation constraint
        </xs:documentation>
      </xs:annotation>
      <xs:element name="element" type="xs:narrowMaxMin"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
```

Content Element Detail (all declarations; 2/2)

[xs:annotation](#) [17]

Type: [anonymous](#) complexType ([extension of xs:openAttrs](#)) [18], complex content
Defined: by reference within ([this](#)) [xs:allModel](#) group; see [XML source](#) [407]

[xs:element](#) [61]

Type: [xs:narrowMaxMin](#) [215], complex content
Defined: locally within ([this](#)) [xs:allModel](#) group; see [XML source](#) [407]

group

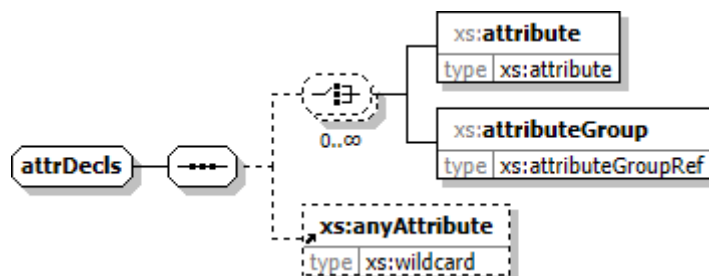
xs:attrDecls

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 3 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [340]

Component Diagram



Complex Content Model

([xs:attribute](#) | [xs:attributeGroup](#))*, [xs:anyAttribute](#)?

Content Model Elements (3):

[xs:anyAttribute](#) [23],

[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34]

[xs:attribute](#) (type [xs:attribute](#)) [29],

Known Usage Locations

- In definitions of other model groups (1):

[xs:complexTypeModel](#) [342]

- In definitions of global complexTypes (7):

[xs:attributeGroup](#) [162], [xs:complexRestrictionType](#) [167], [xs:extensionType](#) [183],
[xs:namedAttributeGroup](#) [209], [xs:restrictionType](#) [229], [xs:simpleExtensionType](#) [236],
[xs:simpleRestrictionType](#) [239]

XML Source (see within schema source: p. 401)

```
<xs:group name="attrDecls">
  <xs:sequence>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:element name="attribute" type="xs:attribute"/>
      <xs:element name="attributeGroup" type="xs:attributeGroupRef"/>
    </xs:choice>
    <xs:element minOccurs="0" ref="xs:anyAttribute"/>
  </xs:sequence>
</xs:group>
```

Content Element Detail (all declarations; 3/3)

[xs:anyAttribute](#) [23]


Type: [xs:wildcard](#) [260], complex content

Defined: by reference within (this) [xs:attrDecls](#) group; see [XML source](#) [401]

[xs:attribute](#) [29]

Type: [xs:attribute](#) [158], complex content

Defined: locally within (this) [xs:attrDecls](#) group; see [XML source](#) [401]

 [xs:attributeGroup](#) [34]

Type: [xs:attributeGroupRef](#) [165], complex content

Defined: locally within ([this](#)) [xs:attrDecls](#) group; see [XML source](#) [401]

group

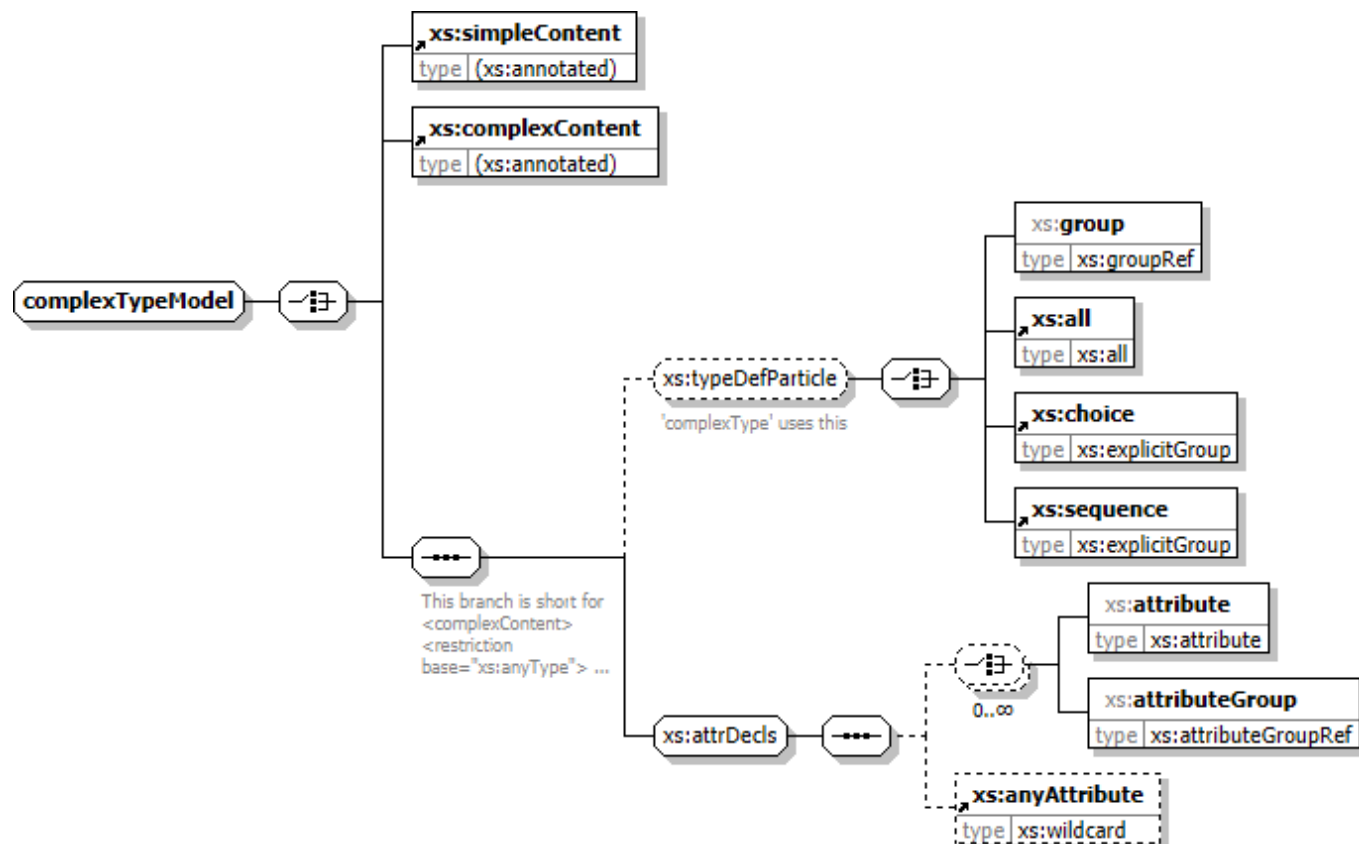
xs:complexTypeModel

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 9 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [342]

Component Diagram



Complex Content Model

[xs:simpleContent](#) | [xs:complexContent](#) | (([xs:group](#) | [xs:all](#) | [xs:choice](#) | [xs:sequence](#))?, ([xs:attribute](#) | [xs:attributeGroup](#))*, [xs:anyAttribute](#)?)

Content Model Elements (9):

[xs:all](#) [12],
[xs:anyAttribute](#) [23],
[xs:attribute](#) (type [xs:attribute](#)) [29],
[xs:attributeGroup](#) (type [xs:attributeGroupRef](#)) [34],
[xs:choice](#) [36],
[xs:complexContent](#) [41],
[xs:group](#) (type [xs:groupRef](#)) [79],
[xs:sequence](#) [128],
[xs:simpleContent](#) [133]

Known Usage Locations

- In definitions of global complexTypes (3):

[xs:complexType](#) [170], [xs:localComplexType](#) [197], [xs:topLevelComplexType](#) [249]

XML Source (see within schema source: p. 401)

```

<xs:group name="complexTypeModel">
  <xs:choice>
    <xs:element ref="xs:simpleContent"/>
  
```

```

<xs:element ref="xs:complexContent"/>
<xs:sequence>
  <xs:annotation>
    <xs:documentation>
      This branch is short for
      &lt;complexContent&gt;
      &lt;restriction base="xs:anyType"&gt;
      ...
      &lt;/restriction&gt;
      &lt;/complexContent&gt;
    </xs:documentation>
  </xs:annotation>
  <xs:group minOccurs="0" ref="xs:typeDefParticle"/>
  <xs:group ref="xs:attrDecls"/>
</xs:sequence>
</xs:choice>
</xs:group>

```

Content Element Detail (all declarations; 9/9)

↔ xs:all [12]

Type: [xs:all](#) [150], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ xs:anyAttribute [23]

Type: [xs:wildcard](#) [260], complex content
Defined: [by reference](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ xs:attribute [29]

Type: [xs:attribute](#) [158], complex content
Defined: [locally](#) [340] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ xs:attributeGroup [34]

Type: [xs:attributeGroupRef](#) [165], complex content
Defined: [locally](#) [341] within [xs:attrDecls](#) group; see [XML source](#) [401]

↔ xs:choice [36]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ xs:complexContent [41]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [42], complex content
Defined: [by reference](#) within ([this](#)) [xs:complexTypeModel](#) group; see [XML source](#) [401]

↔ xs:group [79]

Type: [xs:groupRef](#) [192], complex content
Defined: [locally](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ xs:sequence [128]

Type: [xs:explicitGroup](#) [179], complex content
Defined: [by reference](#) [362] within [xs:typeDefParticle](#) group; see [XML source](#) [400]

↔ xs:simpleContent [133]

Type: [anonymous](#) complexType ([extension of xs:annotated](#)) [134], complex content
Defined: [by reference](#) within ([this](#)) [xs:complexTypeModel](#) group; see [XML source](#) [401]

group

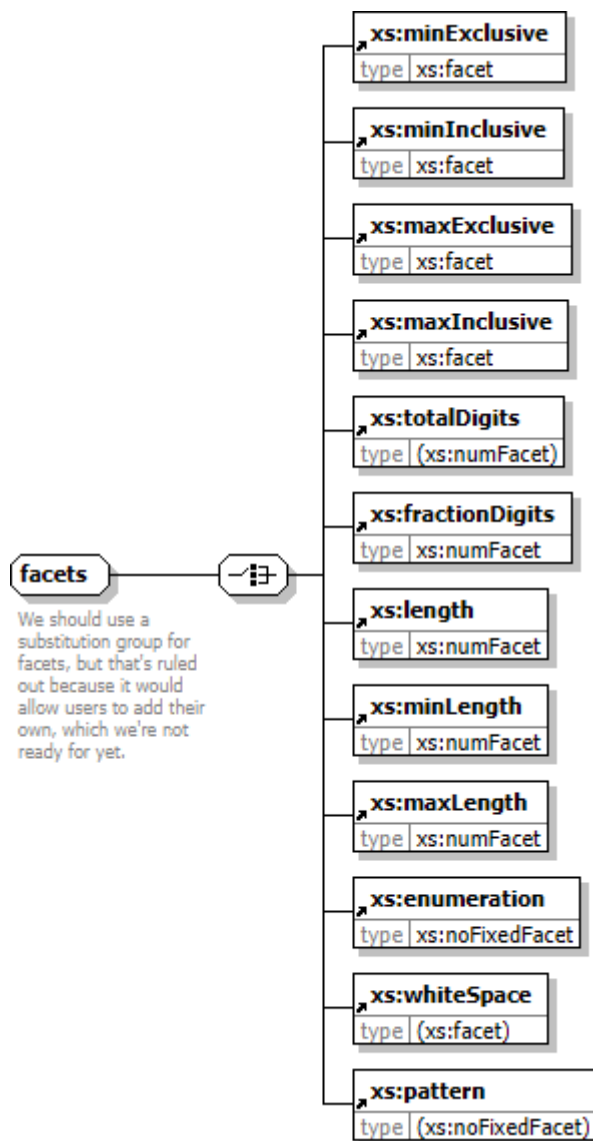
xs:facets

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 12 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [345]

Component Diagram



Complex Content Model

[xs:minExclusive](#) | [xs:minInclusive](#) | [xs:maxExclusive](#) | [xs:maxInclusive](#) | [xs:totalDigits](#) | [xs:fractionDigits](#) | [xs:length](#) | [xs:minLength](#) | [xs:maxLength](#) | [xs:enumeration](#) | [xs:whiteSpace](#) | [xs:pattern](#)

Content Model Elements (12):

[xs:enumeration](#) [65], [xs:fractionDigits](#) [75], [xs:length](#) [90], [xs:maxExclusive](#) [94], [xs:maxInclusive](#) [96], [xs:maxLength](#) [98], [xs:minExclusive](#) [100], [xs:minInclusive](#) [102], [xs:minLength](#) [104], [xs:pattern](#) [109], [xs:totalDigits](#) [140], [xs:whiteSpace](#) [147]

Known Usage Locations

- In definitions of other model groups (1):

[xs:simpleRestrictionModel](#) [358]

Annotation

We should use a substitution group for facets, but that's ruled out because it would allow users to add their own, which we're not ready for yet.


XML Source (see within schema source: p. 424)

```
<xs:group name="facets">
  <xs:annotation>
    <xs:documentation>
      We should use a substitution group for facets, but
      that's ruled out because it would allow users to
      add their own, which we're not ready for yet.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element ref="xs:minExclusive"/>
    <xs:element ref="xs:minInclusive"/>
    <xs:element ref="xs:maxExclusive"/>
    <xs:element ref="xs:maxInclusive"/>
    <xs:element ref="xs:totalDigits"/>
    <xs:element ref="xs:fractionDigits"/>
    <xs:element ref="xs:length"/>
    <xs:element ref="xs:minLength"/>
    <xs:element ref="xs:maxLength"/>
    <xs:element ref="xs:enumeration"/>
    <xs:element ref="xs:whiteSpace"/>
    <xs:element ref="xs:pattern"/>
  </xs:choice>
</xs:group>
```


Content Element Detail (all declarations; 12/12)

 [xs:enumeration](#) [65]

Type: [xs:noFixedFacet](#) [220], complex content
Defined: by reference within [\(this\) xs:facets](#) group; see [XML source](#) [424]

 [xs:fractionDigits](#) [75]


Type: [xs:numFacet](#) [222], complex content
Defined: by reference within [\(this\) xs:facets](#) group; see [XML source](#) [424]

 [xs:length](#) [90]

Type: [xs:numFacet](#) [222], complex content
Defined: by reference within [\(this\) xs:facets](#) group; see [XML source](#) [424]

 [xs:maxExclusive](#) [94]

Type: [xs:facet](#) [186], complex content
Defined: by reference within [\(this\) xs:facets](#) group; see [XML source](#) [424]

 [xs:maxInclusive](#) [96]

Type: [xs:facet](#) [186], complex content
Defined: by reference within [\(this\) xs:facets](#) group; see [XML source](#) [424]

↔ [xs:maxLength](#) [98]

Type: [xs:numFacet](#) [222], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minExclusive](#) [100]

Type: [xs:facet](#) [186], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minInclusive](#) [102]

Type: [xs:facet](#) [186], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minLength](#) [104]

Type: [xs:numFacet](#) [222], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:pattern](#) [109]

Type: [anonymous](#) complexType ([restriction of xs:noFixedFacet](#)) [110], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:totalDigits](#) [140]

Type: [anonymous](#) complexType ([restriction of xs:numFacet](#)) [141], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:whiteSpace](#) [147]

Type: [anonymous](#) complexType ([restriction of xs:facet](#)) [148], complex content
Defined: by reference within ([this](#)) [xs:facets](#) group; see [XML source](#) [424]

group

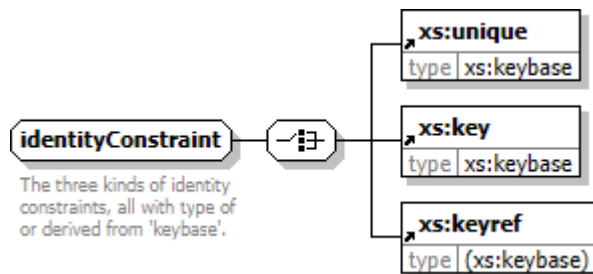
xs:identityConstraint

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 3 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [347]

Component Diagram



Complex Content Model

[xs:unique](#) | [xs:key](#) | [xs:keyref](#)

Content Model Elements (3):

[xs:key](#) [85], [xs:keyref](#) [87], [xs:unique](#) [145]

Known Usage Locations

- In definitions of global complexTypes (4):

[xs:element](#) [174], [xs:localElement](#) [201], [xs:narrowMaxMin](#) [215], [xs:topLevelElement](#) [253]

Annotation

The three kinds of identity constraints, all with type of or derived from 'keybase'.

XML Source (see within schema source: p. 411)

```

<xs:group name="identityConstraint">
  <xs:annotation>
    <xs:documentation>
      The three kinds of identity constraints, all with
      type of or derived from 'keybase'.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element ref="xs:unique"/>
    <xs:element ref="xs:key"/>
    <xs:element ref="xs:keyref"/>
  </xs:choice>
</xs:group>
  
```

Content Element Detail (all declarations; 3/3)

[xs:key](#) [85]

Type: [xs:keybase](#) [195], complex content

Defined: by reference within (this) [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:keyref](#) [87]

Type: [anonymous](#) complexType ([extension of xs:keybase](#)) [88], complex content

Defined: by reference within ([this](#)) [xs:identityConstraint](#) group; see [XML source](#) [412]

↔ [xs:unique](#) [145]

Type: [xs:keybase](#) [195], complex content

Defined: by reference within ([this](#)) [xs:identityConstraint](#) group; see [XML source](#) [411]

group

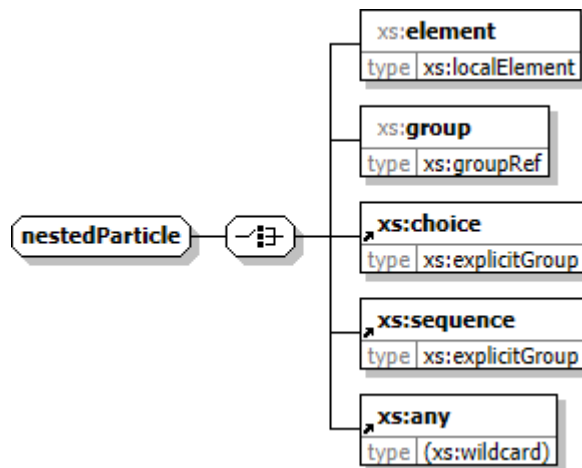
xs:nestedParticle

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 5 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [349]

Component Diagram



Complex Content Model

xs:element | xs:group | xs:choice | xs:sequence | xs:any

Content Model Elements (5):

[xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]
[xs:element](#) (type [xs:localElement](#)) [57],

Known Usage Locations


- In definitions of global complexTypes (2):
[xs:explicitGroup](#) [179], [xs:simpleExplicitGroup](#) [233]

XML Source (see within schema source: p. 400)

```

<xs:group name="nestedParticle">
  <xs:choice>
    <xs:element name="element" type="xs:localElement"/>
    <xs:element name="group" type="xs:groupRef"/>
    <xs:element ref="xs:choice"/>
    <xs:element ref="xs:sequence"/>
    <xs:element ref="xs:any"/>
  </xs:choice>
</xs:group>
  
```

Content Element Detail (all declarations; 5/5)

 [xs:any](#) [20]

Type: [anonymous](#) complexType (extension of [xs:wildcard](#)) [21], complex content
Defined: by reference within (this) [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference within ([this](#)) [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content

Defined: locally within ([this](#)) [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: locally within ([this](#)) [xs:nestedParticle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference within ([this](#)) [xs:nestedParticle](#) group; see [XML source](#) [400]

group

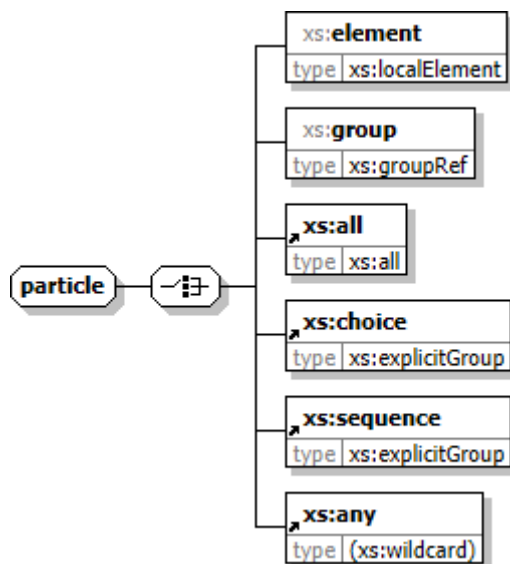
xs:particle

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 6 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [351]

Component Diagram



Complex Content Model

[xs:element](#) | [xs:group](#) | [xs:all](#) | [xs:choice](#) | [xs:sequence](#) | [xs:any](#)

Content Model Elements (6):

[xs:all](#) [12], [xs:element](#) (type [xs:localElement](#)) [57],
[xs:any](#) [20], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]

Known Usage Locations

- In definitions of global complexTypes (1):

[xs:group](#) [188]

XML Source (see within schema source: p. 400)

```

<xs:group name="particle">
  <xs:choice>
    <xs:element name="element" type="xs:localElement"/>
    <xs:element name="group" type="xs:groupRef"/>
    <xs:element ref="xs:all"/>
    <xs:element ref="xs:choice"/>
    <xs:element ref="xs:sequence"/>
    <xs:element ref="xs:any"/>
  </xs:choice>
</xs:group>
  
```

Content Element Detail (all declarations: 6/6)

[xs:all](#) [12]

Type: [xs:all](#) [150], complex content

Defined: by reference within ([this](#)) [xs:particle](#) group; see [XML source](#) [401]

↔ [xs:any](#) [20]

Type: [anonymous](#) complexType ([extension of xs:wildcard](#)) [21], complex content

Defined: by reference within ([this](#)) [xs:particle](#) group; see [XML source](#) [401]

↔ [xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference within ([this](#)) [xs:particle](#) group; see [XML source](#) [401]

↔ [xs:element](#) [57]

Type: [xs:localElement](#) [201], complex content

Defined: locally within ([this](#)) [xs:particle](#) group; see [XML source](#) [400]

↔ [xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: locally within ([this](#)) [xs:particle](#) group; see [XML source](#) [400]

↔ [xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference within ([this](#)) [xs:particle](#) group; see [XML source](#) [401]

group

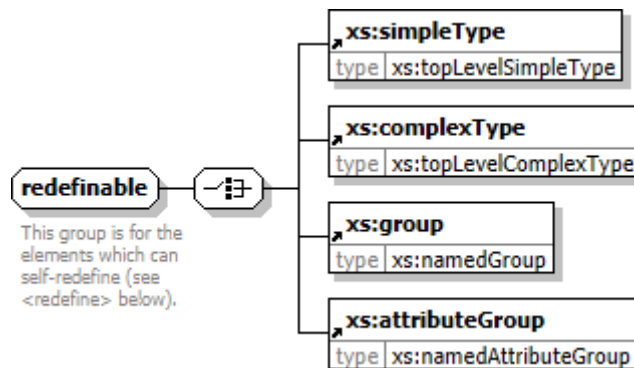
xs:redefinable

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 4 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [353]

Component Diagram



Complex Content Model

[xs:simpleType](#) | [xs:complexType](#) | [xs:group](#) | [xs:attributeGroup](#)

Content Model Elements (4):

[xs:attributeGroup](#) [32], [xs:complexType](#) [44], [xs:group](#) [77], [xs:simpleType](#) [135]

Known Usage Locations

- In definitions of other model groups (1):
[xs:schemaTop](#) [355]
- In definitions of anonymous complexTypes of elements (1):
[xs:redefine](#) [111]

Annotation

This group is for the elements which can self-redefine (see `<redefine>` below).

XML Source (see within schema source: p. 398)

```

<xs:group name="redefinable">
  <xs:annotation>
    <xs:documentation>
      This group is for the
      elements which can self-redefine (see &lt;redefine&gt; below).
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element ref="xs:simpleType"/>
    <xs:element ref="xs:complexType"/>
    <xs:element ref="xs:group"/>
    <xs:element ref="xs:attributeGroup"/>
  </xs:choice>
</xs:group>
  
```

Content Element Detail (all declarations; 4/4)

[xs:attributeGroup](#) [32]

Type: [xs:namedAttributeGroup](#) [209], complex content
Defined: by reference within ([this](#)) [xs:redefinable](#) group; see [XML source](#) [398]

[xs:complexType](#) [44]

Type: [xs:topLevelComplexType](#) [249], complex content
Defined: by reference within ([this](#)) [xs:redefinable](#) group; see [XML source](#) [398]

[xs:group](#) [77]

Type: [xs:namedGroup](#) [212], complex content
Defined: by reference within ([this](#)) [xs:redefinable](#) group; see [XML source](#) [398]

[xs:simpleType](#) [135]

Type: [xs:topLevelSimpleType](#) [257], complex content
Defined: by reference within ([this](#)) [xs:redefinable](#) group; see [XML source](#) [398]

group

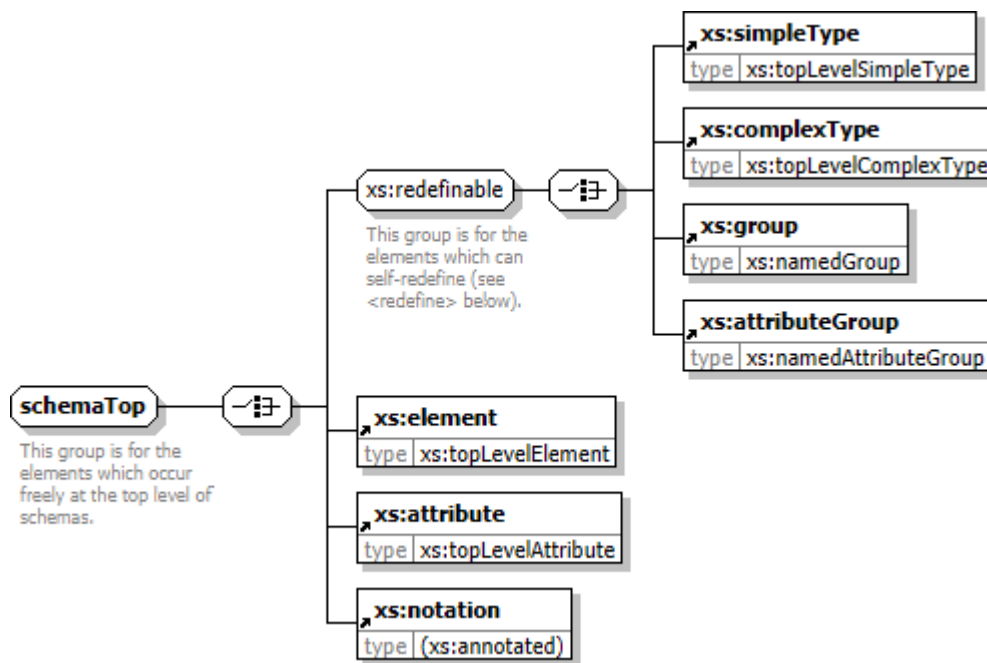
xs:schemaTop

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 7 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [355]

Component Diagram



Complex Content Model

[xs:simpleType](#) | [xs:complexType](#) | [xs:group](#) | [xs:attributeGroup](#) | [xs:element](#) | [xs:attribute](#) | [xs:notation](#)

Content Model Elements (7):

[xs:attribute](#) [27], [xs:attributeGroup](#) [32], [xs:complexType](#) [44], [xs:element](#) [53], [xs:group](#) [77],
[xs:notation](#) [106], [xs:simpleType](#) [135]

Known Usage Locations

- In definitions of anonymous complexTypes of elements (1):

[xs:schema](#) [7]

Annotation

This group is for the elements which occur freely at the top level of schemas. All of their types are based on the "annotated" type by extension.

XML Source (see within schema source: p. 398)

```
<xs:group name="schemaTop">
  <xs:annotation>
    <xs:documentation>
      This group is for the
      elements which occur freely at the top level of schemas.
      All of their types are based on the "annotated" type by extension.
    </xs:documentation>
  </xs:annotation>
```

```
<xs:choice>
  <xs:group ref="xs:redefinable"/>
  <xs:element ref="xs:element"/>
  <xs:element ref="xs:attribute"/>
  <xs:element ref="xs:notation"/>
</xs:choice>
</xs:group>
```

Content Element Detail (all declarations; 7/7)

xs:attribute [27]

Type: [xs:topLevelAttribute](#) [246], complex content
Defined: by reference within ([this](#)) [xs:schemaTop](#) group; see [XML source](#) [398]

xs:attributeGroup [32]

Type: [xs:namedAttributeGroup](#) [209], complex content
Defined: by reference [354] within [xs:redefinable](#) group; see [XML source](#) [398]

xs:complexType [44]

Type: [xs:topLevelComplexType](#) [249], complex content
Defined: by reference [354] within [xs:redefinable](#) group; see [XML source](#) [398]

xs:element [53]

Type: [xs:topLevelElement](#) [253], complex content
Defined: by reference within ([this](#)) [xs:schemaTop](#) group; see [XML source](#) [398]

xs:group [77]

Type: [xs:namedGroup](#) [212], complex content
Defined: by reference [354] within [xs:redefinable](#) group; see [XML source](#) [398]

xs:notation [106]

Type: [anonymous complexType](#) ([extension of xs:annotated](#)) [107], complex content
Defined: by reference within ([this](#)) [xs:schemaTop](#) group; see [XML source](#) [398]

xs:simpleType [135]

Type: [xs:topLevelSimpleType](#) [257], complex content
Defined: by reference [354] within [xs:redefinable](#) group; see [XML source](#) [398]

group

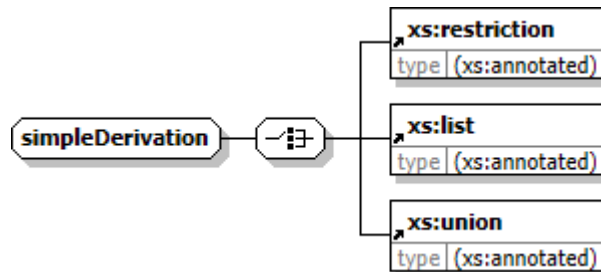
xs:simpleDerivation

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 3 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [357]

Component Diagram



Complex Content Model

[xs:restriction](#) | [xs:list](#) | [xs:union](#)

Content Model Elements (3):

[xs:list](#) [92], [xs:restriction](#) [114], [xs:union](#) [142]

Known Usage Locations

- In definitions of global complexTypes (3):

[xs:localSimpleType](#) [206], [xs:simpleType](#) [243], [xs:topLevelSimpleType](#) [257]

XML Source (see within schema source: p. 423)

```

<xs:group name="simpleDerivation">
  <xs:choice>
    <xs:element ref="xs:restriction"/>
    <xs:element ref="xs:list"/>
    <xs:element ref="xs:union"/>
  </xs:choice>
</xs:group>
  
```

Content Element Detail (all declarations; 3/3)

[xs:list](#) [92]

Type: anonymous complexType (extension of [xs:annotated](#)) [93], complex content
Defined: by reference within (this) [xs:simpleDerivation](#) group; see [XML source](#) [423]

[xs:restriction](#) [114]

Type: anonymous complexType (extension of [xs:annotated](#)) [115], complex content
Defined: by reference within (this) [xs:simpleDerivation](#) group; see [XML source](#) [423]

[xs:union](#) [142]

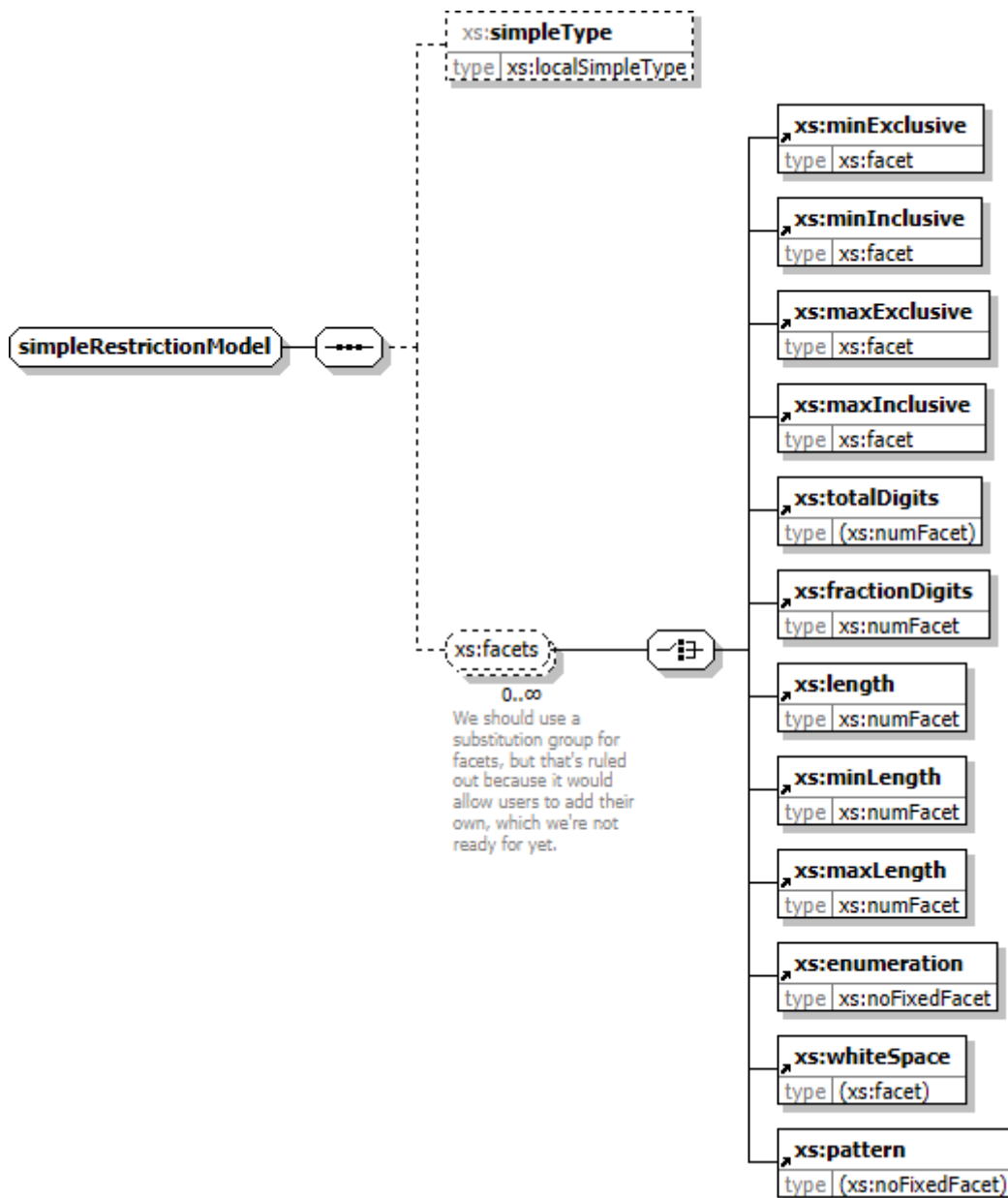
Type: anonymous complexType (extension of [xs:annotated](#)) [143], complex content
Defined: by reference within (this) [xs:simpleDerivation](#) group; see [XML source](#) [423]

group

xs:simpleRestrictionModel

Namespace: <http://www.w3.org/2001/XMLSchema>
 Content: 13 elements
 Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [359]

Component Diagram



Complex Content Model

xs:simpleType?, (xs:minExclusive | xs:minInclusive | xs:maxExclusive | xs:maxInclusive | xs:totalDigits | xs:fractionDigits | xs:length | xs:minLength | xs:maxLength | xs:enumeration | xs:whiteSpace | xs:pattern)*

Content Model Elements (13):

- xs:enumeration [65],
- xs:fractionDigits [75],
- xs:length [90],
- xs:maxExclusive [94],
- xs:maxInclusive [96],
- xs:minInclusive [102],
- xs:minLength [104],
- xs:pattern [109],
- xs:simpleType (type xs:localSimpleType) [138],
- xs:totalDigits [140],

[xs:maxLength](#) [98], [xs:whiteSpace](#) [147]
[xs:minExclusive](#) [100],

Known Usage Locations

- In definitions of global complexTypes (2):
[xs:restrictionType](#) [229], [xs:simpleRestrictionType](#) [239]
- In definitions of anonymous complexTypes of elements (1):
[xs:restriction](#) [114]


XML Source (see within schema source: p. 424)

```
<xs:group name="simpleRestrictionModel">  
  <xs:sequence>  
    <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>  
    <xs:group minOccurs="0" maxOccurs="unbounded" minOccurs="0" ref="xs:facets"/>  
  </xs:sequence>  
</xs:group>
```


Content Element Detail (all declarations; 13/13)

 [xs:enumeration](#) [65]

Type: [xs:noFixedFacet](#) [220], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:fractionDigits](#) [75]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:length](#) [90]


Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:maxExclusive](#) [94]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:maxInclusive](#) [96]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [345] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:maxLength](#) [98]

Type: [xs:numFacet](#) [222], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:minExclusive](#) [100]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

 [xs:minInclusive](#) [102]

Type: [xs:facet](#) [186], complex content
Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:minLength](#) [104]

Type: [xs:numFacet](#) [222], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:pattern](#) [109]

Type: [anonymous](#) complexType ([restriction of xs:noFixedFacet](#)) [110], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:simpleType](#) [138]

Type: [xs:localSimpleType](#) [206], complex content

Defined: locally within ([this](#)) [xs:simpleRestrictionModel](#) group; see [XML source](#) [424]

↔ [xs:totalDigits](#) [140]

Type: [anonymous](#) complexType ([restriction of xs:numFacet](#)) [141], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

↔ [xs:whiteSpace](#) [147]

Type: [anonymous](#) complexType ([restriction of xs:facet](#)) [148], complex content

Defined: [by reference](#) [346] within [xs:facets](#) group; see [XML source](#) [424]

group

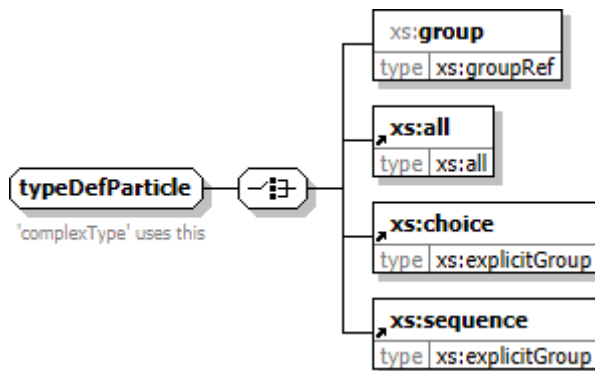
xs:typeDefParticle

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 4 elements

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [361]

Component Diagram



Complex Content Model

[xs:group](#) | [xs:all](#) | [xs:choice](#) | [xs:sequence](#)

Content Model Elements (4):

[xs:all](#) [12], [xs:group](#) (type [xs:groupRef](#)) [79],
[xs:choice](#) [36], [xs:sequence](#) [128]

Known Usage Locations

- In definitions of other model groups (1):
[xs:complexTypeModel](#) [342]
- In definitions of global complexTypes (3):
[xs:complexRestrictionType](#) [167], [xs:extensionType](#) [183], [xs:restrictionType](#) [229]

Annotation

'complexType' uses this

XML Source (see within schema source: p. 400)

```
<xs:group name="typeDefParticle">
  <xs:annotation>
    <xs:documentation>
      'complexType' uses this
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element name="group" type="xs:groupRef"/>
    <xs:element ref="xs:all"/>
    <xs:element ref="xs:choice"/>
    <xs:element ref="xs:sequence"/>
  </xs:choice>
</xs:group>
```

Content Element Detail (all declarations; 4/4)

[xs:all](#) [12]

Type: [xs:all](#) [150], complex content

Defined: by reference within ([this](#)) [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:choice](#) [36]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference within ([this](#)) [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:group](#) [79]

Type: [xs:groupRef](#) [192], complex content

Defined: locally within ([this](#)) [xs:typeDefParticle](#) group; see [XML source](#) [400]

[xs:sequence](#) [128]

Type: [xs:explicitGroup](#) [179], complex content

Defined: by reference within ([this](#)) [xs:typeDefParticle](#) group; see [XML source](#) [400]

Attribute Groups

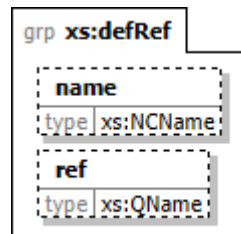
attributeGroup xs:defRef

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 2 attributes

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [363]

Component Diagram



for element, group and attributeGroup, which both define and reference

XML Representation Summary

```

<...
  name = xs:NCName
  ref = xs:QName
...>
  
```

Known Usage Locations

- In definitions of global complexTypes (4):

[xs:attribute](#) [158], [xs:attributeGroup](#) [162], [xs:element](#) [174], [xs:group](#) [188]

Annotation

for element, group and attributeGroup,
which both define and reference

XML Source (see within schema source: p. 400)

```

<xs:attributeGroup name="defRef">
  <xs:annotation>
    <xs:documentation>
      for element, group and attributeGroup,
      which both define and reference
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="name" type="xs:NCName" />
  <xs:attribute name="ref" type="xs:QName" />
</xs:attributeGroup>
  
```

Attribute Detail (all declarations; 2/2)

■ name

Type: [xs:NCName](#) [308]

Use: optional

Defined: locally within ([this](#)) [xs:defRef](#) attributeGroup; see [XML source](#) [400]

■ ref

Type: [xs:QName](#) [322]

Use: optional

Defined: locally within ([this](#)) [xs:defRef](#) attributeGroup; see [XML source](#) [400]

attributeGroup

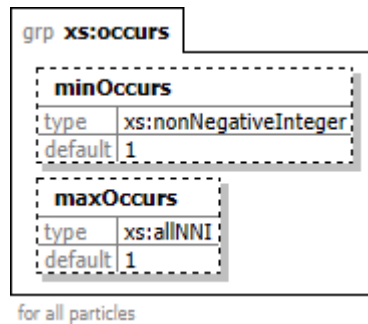
XS:OCCURS

Namespace: <http://www.w3.org/2001/XMLSchema>

Content: 2 attributes

Defined: globally in [XMLSchema.xsd](#); see [XML source](#) [365]

Component Diagram



XML Representation Summary

```

<...
  minOccurs = xs:nonNegativeInteger : "1"
  maxOccurs = (xs:nonNegativeInteger | "unbounded") : "1"
...>
  
```

Known Usage Locations

- In definitions of global complexTypes (2):
[xs:element](#) [174], [xs:group](#) [188]
- In definitions of anonymous complexTypes of elements (1):
[xs:any](#) [20]

Annotation

for all particles

XML Source (see within schema source: p. 400)

```

<xs:attributeGroup name="occurs">
  <xs:annotation>
    <xs:documentation>
      for all particles
    </xs:documentation>
  </xs:annotation>
  <xs:attribute default="1" name="minOccurs" type="xs:nonNegativeInteger" use="optional"/>
  <xs:attribute default="1" name="maxOccurs" type="xs:allNNI" use="optional"/>
</xs:attributeGroup>
  
```

Attribute Detail (all declarations; 2/2)

■ maxOccurs

Type: [xs:allNNI](#) [263]

Use: optional

Defined: locally within (this) [xs:occurs](#) attributeGroup; see [XML source](#) [400]

Attribute Value

`xs:nonNegativeInteger` | "unbounded"

Default: "1"

■ minOccurs

Type: `xs:nonNegativeInteger` [314]

Use: optional

Defined: locally within ([this](#)) `xs:occurs` attributeGroup; see [XML source](#) [400]

Attribute Value

Default: "1"

Namespace {http://www.w3.org/XML/1998/namespace}

Global Attributes

attribute

xml:base

Namespace: <http://www.w3.org/XML/1998/namespace>

Type: [xs:anyURI](#) [265]

Defined: globally in [xml.xsd](#); see [XML source](#) [368]

XML Representation Summary

```
<...  
  xml:base = xs:anyURI  
...>
```

Attribute Value Detail:

WhiteSpace: collapse

Known Usage Locations

- In definitions of attributeGroups (1):

[xml:specialAttrs](#) [374]

Annotation

base (as an attribute name)

denotes an attribute whose value provides a URI to be used as the base for interpreting any relative URIs in the scope of the element on which it appears; its value is inherited. This name is reserved by virtue of its definition in the XML Base specification.

See <http://www.w3.org/TR/xmlbase/> for information about this attribute.

XML Source (see within schema source: p. 379)

```
<xs:attribute name="base" type="xs:anyURI">  
  <xs:annotation>  
    <xs:documentation>  
      <div>  
        <h3>base \(as an attribute name\)</h3>  
        <p>  
          denotes an attribute whose value  
          provides a URI to be used as the base for interpreting any  
          relative URIs in the scope of the element on which it  
          appears; its value is inherited. This name is reserved  
          by virtue of its definition in the XML Base specification.  
        </p>  
        <p>  
          See  
          <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a>  
          for information about this attribute.  
        </p>  
      </div>  
    </xs:documentation>  
  </xs:annotation>  
</xs:attribute>
```

attribute

xml:id

Namespace: <http://www.w3.org/XML/1998/namespace>

Type: [xs:ID](#) [295]

Defined: globally in [xml.xsd](#); see [XML source](#) [369]

XML Representation Summary

```
<...
  xml:id = xs:ID
...>
```

Known Usage Locations

- In definitions of [attributeGroups](#) (1):
[xml:specialAttrs](#) [375]

Annotation

id (as an attribute name)

denotes an attribute whose value should be interpreted as if declared to be of type ID. This name is reserved by virtue of its definition in the [xml:id](#) specification.

See <http://www.w3.org/TR/xml-id/> for information about this attribute.

XML Source (see within schema source: p. 379)

```
<xs:attribute name="id" type="xs:ID">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h3>id (as an attribute name)</h3>
        <p>
          denotes an attribute whose value
          should be interpreted as if declared to be of type ID.
          This name is reserved by virtue of its definition in the
          xml:id specification.
        </p>
        <p>
          See
          <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a>
          for information about this attribute.
        </p>
      </div>
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
```

attribute

xml:lang

Namespace: <http://www.w3.org/XML/1998/namespace>

Type: [anonymous simpleType](#) (union of ([xs:language](#) | [restriction of xs:string](#))) [370]

Defined: globally in [xml.xsd](#); see [XML source](#) [371]

XML Representation Summary

```
<...
  xml:lang = (xs:language | "")
...>
```

Included in content model of elements (2):

[xs:documentation](#) [51], [xs:schema](#) [7]

Known Usage Locations

- In definitions of [attributeGroups](#) (1):

[xml:specialAttrs](#) [375]

- In definitions of [anonymous complexTypes](#) of elements (2):

[xs:documentation](#) [52], [xs:schema](#) [10]

Annotation

lang (as an attribute name)

denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.

Notes

Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.

See BCP 47 at <http://www.rfc-editor.org/rfc/bcp/bcp47.txt> and the IANA language subtag registry at <http://www.iana.org/assignments/language-subtag-registry> for further information.

The union allows for the 'un-declaration' of xml:lang with the empty string.

Anonymous Type Detail

Type Derivation Tree

```
union of (xs:language | restriction of xs:string)
└─ simpleType
```

Derivation: by union

Member Types

1. [xs:language](#)

2. anonymous simpleType:

Derivation: [restriction of xs:string](#)

Facets: enumeration: ""

XML Source (see within schema source: p. 378)

```

<xs:attribute name="lang">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h3>lang (as an attribute name)</h3>
        <p>
          denotes an attribute whose value
          is a language code for the natural language of the content of
          any element; its value is inherited. This name is reserved
          by virtue of its definition in the XML specification.
        </p>
      </div>
      <div>
        <h4>Notes</h4>
        <p>
          Attempting to install the relevant ISO 2- and 3-letter
          codes as the enumerated possible values is probably never
          going to be a realistic possibility.
        </p>
        <p>
          See BCP 47 at
          <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">
            http://www.rfc-editor.org/rfc/bcp/bcp47.txt
          </a>
          and the IANA language subtag registry at
          <a href="http://www.iana.org/assignments/language-subtag-registry">
            http://www.iana.org/assignments/language-subtag-registry
          </a>
          for further information.
        </p>
        <p>
          The union allows for the 'un-declaration' of xml:lang with
          the empty string.
        </p>
      </div>
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:union memberTypes="xs:language">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value=""/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</xs:attribute>

```

attribute

xml:space

Namespace: <http://www.w3.org/XML/1998/namespace>
Type: [anonymous simpleType](#) ([restriction of xs:NCName](#)) [372]
Defined: globally in [xml.xsd](#); see [XML source](#) [372]

XML Representation Summary

```
<...
  xml:space = enumeration of xs:NCName
...>
```

Attribute Value Detail:

Enumeration: "default", "preserve"

Known Usage Locations

- In definitions of [attributeGroups](#) (1):
[xml:specialAttrs](#) [375]

Annotation

space (as an attribute name)

denotes an attribute whose value is a keyword indicating what whitespace processing discipline is intended for the content of the element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.

Anonymous Type Detail

Type Derivation Tree

```
xs:anySimpleType (restriction)
├── xs:string [328] (restriction)
│   └── xs:normalizedString [317] (restriction)
│       └── xs:token [331] (restriction)
│           └── xs:Name [304] (restriction)
│               └── xs:NCName [308] (restriction)
│                   └── simpleType
```

Derivation: [restriction of xs:NCName](#)
Facets: [enumeration:](#) "default", "preserve"

XML Source (see within schema source: p. 379)

```
<xs:attribute name="space">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h3>space (as an attribute name)</h3>
        <p>
          denotes an attribute whose
          value is a keyword indicating what whitespace processing
          discipline is intended for the content of the element; its
          value is inherited. This name is reserved by virtue of its
          definition in the XML specification.
        </p>
      </div>
    </xs:documentation>
  </xs:annotation>
```



```
<xs:simpleType>  
  <xs:restriction base="xs:NCName">  
    <xs:enumeration value="default"/>  
    <xs:enumeration value="preserve"/>  
  </xs:restriction>  
</xs:simpleType>  
</xs:attribute>
```

Attribute Groups

attributeGroup

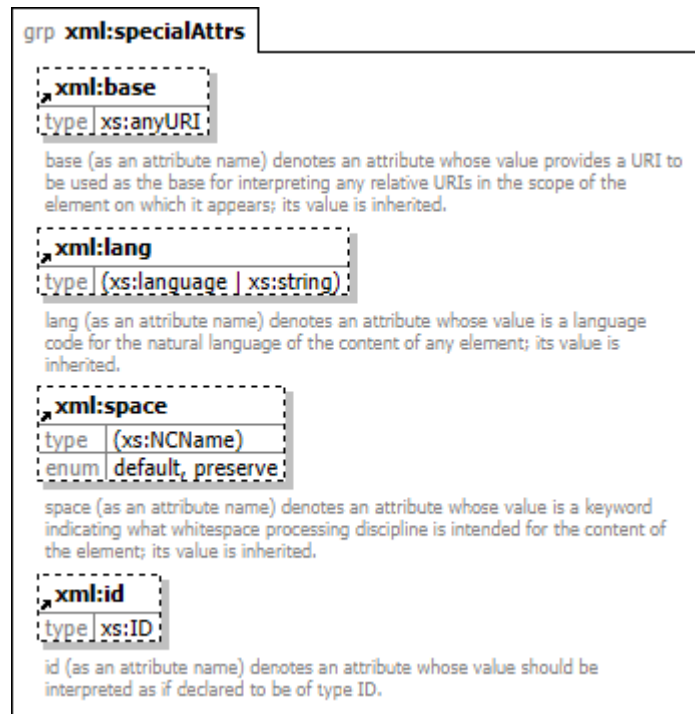
xml:specialAttrs

Namespace: <http://www.w3.org/XML/1998/namespace>

Content: 4 [attributes](#)

Defined: globally in [xml.xsd](#); see [XML source](#) [374]

Component Diagram



XML Representation Summary

```
<...
  xml:base = xs:anyURI
  xml:lang = (xs:language | "")
  xml:space = ("default" | "preserve")
  xml:id = xs:ID
...>
```

XML Source (see within schema source: p. 380)

```
<xs:attributeGroup name="specialAttrs">
  <xs:attribute ref="xml:base"/>
  <xs:attribute ref="xml:lang"/>
  <xs:attribute ref="xml:space"/>
  <xs:attribute ref="xml:id"/>
</xs:attributeGroup>
```

Attribute Detail (all declarations; 4/4)

■ [xml:base](#) [368]

Type: [xs:anyURI](#) [265]

Use: optional

Defined: by reference within ([this](#)) [xml:specialAttrs](#) attributeGroup; see [XML source](#) [380]

■ `xml:id` [369]

Type: `xs:ID` [295]

Use: optional

Defined: by reference within (this) `xml:specialAttrs` attributeGroup; see [XML source](#) [380]

■ `xml:lang` [370]

Type: anonymous simpleType (union of (`xs:language` | restriction of `xs:string`)) [370]

Use: optional

Defined: by reference within (this) `xml:specialAttrs` attributeGroup; see [XML source](#) [380]

Attribute Value

```
xs:language | ""
```

■ `xml:space` [372]

Type: anonymous simpleType (restriction of `xs:NCName`) [372]

Use: optional

Defined: by reference within (this) `xml:specialAttrs` attributeGroup; see [XML source](#) [380]

Attribute Value

```
enumeration of xs:NCName
```

Enumeration: "default", "preserve"

Schema "xml.xsd"

Schema Source:

<http://www.w3.org/2001/xml.xsd>; see [XML source](#) [378]

Target Namespace:

<http://www.w3.org/XML/1998/namespace>

Defined Components:

[global attributes](#) (4), [attribute groups](#) (1)

Default Namespace-Qualified Form:

Local Elements: unqualified; Local Attributes: unqualified

Imported by Schemas (1):

[XMLSchema.xsd](#)

Annotation

About the XML namespace

This schema document describes the XML namespace, in a form suitable for import by other schema documents.

See <http://www.w3.org/XML/1998/namespace.html> and <http://www.w3.org/TR/REC-xml> for information about this namespace.

Note that local names in this namespace are intended to be defined only by the World Wide Web Consortium or its subgroups. The names currently defined in this namespace are listed below. They should not be used with conflicting semantics by any Working Group, specification, or document instance.

See further below in this document for more information about [how to refer to this schema document from your own XSD schema documents](#) and about [the namespace-versioning policy governing this schema document](#).

Father (in any context at all)

denotes Jon Bosak, the chair of the original XML Working Group. This name is reserved by the following decision of the W3C XML Plenary and XML Coordination groups:

In appreciation for his vision, leadership and dedication the W3C XML Plenary on this 10th day of February, 2000, reserves for Jon Bosak in perpetuity the XML name "xml:Father".

About this schema document

This schema defines attributes and an attribute group suitable for use by schemas wishing to allow `xml:base`, `xml:lang`, `xml:space` or `xml:id` attributes on elements they define.

To enable this, such a schema must import this schema for the XML namespace, e.g. as follows:

```
<schema . . .>
. . .
<import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
```

or

```
<import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
```

Subsequently, qualified reference to any of the attributes or the group defined below will have the desired effect, e.g.

```
<type . . . >
.
.
<attributeGroup ref="xml:specialAttrs"/>
```

will define a type which will schema-validate an instance element with any of those attributes.

Versioning policy for this schema document





In keeping with the XML Schema WG's standard versioning policy, this schema document will persist at <http://www.w3.org/2009/01/xml.xsd>.


At the date of issue it can also be found at <http://www.w3.org/2001/xml.xsd>.

The schema document at that URI may however change in the future, in order to remain compatible with the latest version of XML Schema itself, or with the XML namespace itself. In other words, if the XML Schema or XML namespaces change, the version of this document at <http://www.w3.org/2001/xml.xsd> will change accordingly; the version at <http://www.w3.org/2009/01/xml.xsd> will not change.

Previous dated (and unchanging) versions of this schema document are at:

- <http://www.w3.org/2009/01/xml.xsd>
- <http://www.w3.org/2007/08/xml.xsd>
- <http://www.w3.org/2004/10/xml.xsd>
- <http://www.w3.org/2001/03/xml.xsd>

Global Attribute Summary		Page
 xml:base	base (as an attribute name) denotes an attribute whose value provides a URI to be used as the base for interpreting any relative URIs in the scope of the element on which it appears; its value is inherited. Type: xs:anyURI [265] Defined: globally; see XML source [368] Used: at 1 location	368
 xml:id	id (as an attribute name) denotes an attribute whose value should be interpreted as if declared to be of type ID. Type: xs:ID [295] Defined: globally; see XML source [369] Used: at 1 location	369
 xml:lang	lang (as an attribute name) denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. Type: anonymous simpleType (union of (xs:language restriction of xs:string)) [370] Defined: globally; see XML source [371] Used: at 3 locations	370
 xml:space	space (as an attribute name) denotes an attribute whose value is a keyword indicating what whitespace processing discipline is intended for the content of the element; its value is inherited. Type: anonymous simpleType (restriction of xs:NCName) [372] Defined: globally; see XML source [372] Used: at 1 location	372

Attribute Group Summary		Page
 xml:specialAttrs	Content: 4 attributes Defined: globally; see XML source [374] Includes: definitions of 4 attributes Used: never	374

Schema XML Source

```

<?xml version="1.0"?>
<?xml-stylesheet href="../2008/09/xsd.xsl" type="text/xsl"?>
<xs:schema targetNamespace="http://www.w3.org/XML/1998/namespace" xml:lang="en"
xmlns="http://www.w3.org/1999/xhtml" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h1>About the XML namespace</h1>
        <div class="bodytext">
          <p>
            This schema document describes the XML namespace, in a form
            suitable for import by other schema documents.
          </p>
          <p>
            See
            <a href="http://www.w3.org/XML/1998/namespace.html">
              http://www.w3.org/XML/1998/namespace.html
            </a>
            and
            <a href="http://www.w3.org/TR/REC-xml">
              http://www.w3.org/TR/REC-xml
            </a>
            for information
            about this namespace.
          </p>
          <p>
            Note that local names in this namespace are intended to be
            defined only by the World Wide Web Consortium or its subgroups.
            The names currently defined in this namespace are listed below.
            They should not be used with conflicting semantics by any Working
            Group, specification, or document instance.
          </p>
          <p>
            See further below in this document for more information about
            <a href="#usage">
              how to refer to this schema document from your own
              XSD schema documents
            </a>
            and about
            <a href="#nsversioning">
              the
              namespace-versioning policy governing this schema document
            </a>
          </p>
        </div>
      </div>
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="lang">
    <xs:annotation>
      <xs:documentation>
        <div>
          <h3>lang (as an attribute name)</h3>
          <p>
            denotes an attribute whose value
            is a language code for the natural language of the content of
            any element; its value is inherited. This name is reserved
            by virtue of its definition in the XML specification.
          </p>
        </div>
        <div>
          <h4>Notes</h4>
          <p>
            Attempting to install the relevant ISO 2- and 3-letter
            codes as the enumerated possible values is probably never
            going to be a realistic possibility.
          </p>
          <p>
            See BCP 47 at
            <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">
              http://www.rfc-editor.org/rfc/bcp/bcp47.txt
            </a>
            and the IANA language subtag registry at
            <a href="http://www.iana.org/assignments/language-subtag-registry">

```

```

    http://www.iana.org/assignments/language-subtag-registry
  </a>
  for further information.
</p>
<p>
  The union allows for the 'un-declaration' of xml:lang with
  the empty string.
</p>
</div>
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:union memberTypes="xs:language">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value=""/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="space">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h3>space (as an attribute name)</h3>
        <p>
          denotes an attribute whose
          value is a keyword indicating what whitespace processing
          discipline is intended for the content of the element; its
          value is inherited. This name is reserved by virtue of its
          definition in the XML specification.
        </p>
      </div>
    </xs:documentation>
  </xs:annotation>
</xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="default"/>
    <xs:enumeration value="preserve"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="base" type="xs:anyURI">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h3>base (as an attribute name)</h3>
        <p>
          denotes an attribute whose value
          provides a URI to be used as the base for interpreting any
          relative URIs in the scope of the element on which it
          appears; its value is inherited. This name is reserved
          by virtue of its definition in the XML Base specification.
        </p>
        <p>
          See
          <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a>
          for information about this attribute.
        </p>
      </div>
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="id" type="xs:ID">
  <xs:annotation>
    <xs:documentation>
      <div>
        <h3>id (as an attribute name)</h3>
        <p>
          denotes an attribute whose value
          should be interpreted as if declared to be of type ID.
          This name is reserved by virtue of its definition in the
          xml:id specification.
        </p>
        <p>
          See

```

```

    <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a>
    for information about this attribute.
  </p>
</div>
</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attributeGroup name="specialAttrs">
  <xs:attribute ref="xml:base"/>
  <xs:attribute ref="xml:lang"/>
  <xs:attribute ref="xml:space"/>
  <xs:attribute ref="xml:id"/>
</xs:attributeGroup>
<xs:annotation>
  <xs:documentation>
    <div>
      <h3>Father (in any context at all)</h3>
      <div class="bodytext">
        <p>
          denotes Jon Bosak, the chair of
          the original XML Working Group. This name is reserved by
          the following decision of the W3C XML Plenary and
          XML Coordination groups:
        </p>
        <blockquote>
          <p>
            In appreciation for his vision, leadership and
            dedication the W3C XML Plenary on this 10th day of
            February, 2000, reserves for Jon Bosak in perpetuity
            the XML name "xml:Father".
          </p>
        </blockquote>
      </div>
    </div>
  </xs:documentation>
</xs:annotation>
<xs:annotation>
  <xs:documentation>
    <div id="usage" xml:id="usage">
      <h2>
        <a name="usage">About this schema document</a>
      </h2>
      <div class="bodytext">
        <p>
          This schema defines attributes and an attribute group suitable
          for use by schemas wishing to allow
          <code>xml:base</code>
          ,
          <code>xml:lang</code>
          ,
          <code>xml:space</code>
          or
          <code>xml:id</code>
          attributes on elements they define.
        </p>
        <p>
          To enable this, such a schema must import this schema for
          the XML namespace, e.g. as follows:
        </p>
        <pre>
          &lt;schema . . .&gt;
            . . .
            &lt;import namespace="http://www.w3.org/XML/1998/namespace"
              schemaLocation="http://www.w3.org/2001/xml.xsd"/&gt;
        </pre>
        <p>
          or
        </p>
        <pre>
          &lt;import namespace="http://www.w3.org/XML/1998/namespace"
            schemaLocation="http://www.w3.org/2009/01/xml.xsd"/&gt;
        </pre>
        <p>
          Subsequently, qualified reference to any of the attributes or the
          group defined below will have the desired effect, e.g.
        </p>
        <pre>

```



```

    &lt;type . . .&gt;
    . . .
    &lt;attributeGroup ref="xml:specialAttrs"/&gt;
  </pre>
  <p>
    will define a type which will schema-validate an instance element
    with any of those attributes.
  </p>
</div>
</xs:documentation>
</xs:annotation>
<xs:annotation>
  <xs:documentation>
    <div id="nsversioning" xml:id="nsversioning">
      <h2>
        <a name="nsversioning">Versioning policy for this schema document</a>
      </h2>
      <div class="bodytext">
        <p>
          In keeping with the XML Schema WG's standard versioning
          policy, this schema document will persist at
          <a href="http://www.w3.org/2009/01/xml.xsd">
            http://www.w3.org/2009/01/xml.xsd
          </a>
          .
        </p>
        <p>
          At the date of issue it can also be found at
          <a href="http://www.w3.org/2001/xml.xsd">
            http://www.w3.org/2001/xml.xsd
          </a>
          .
        </p>
        <p>
          The schema document at that URI may however change in the future,
          in order to remain compatible with the latest version of XML
          Schema itself, or with the XML namespace itself. In other words,
          if the XML Schema or XML namespaces change, the version of this
          document at
          <a href="http://www.w3.org/2001/xml.xsd">
            http://www.w3.org/2001/xml.xsd
          </a>
          will change accordingly; the version at
          <a href="http://www.w3.org/2009/01/xml.xsd">
            http://www.w3.org/2009/01/xml.xsd
          </a>
          will not change.
        </p>
        <p>
          Previous dated (and unchanging) versions of this schema
          document are at:
        </p>
        <ul>
          <li>
            <a href="http://www.w3.org/2009/01/xml.xsd">
              http://www.w3.org/2009/01/xml.xsd
            </a>
          </li>
          <li>
            <a href="http://www.w3.org/2007/08/xml.xsd">
              http://www.w3.org/2007/08/xml.xsd
            </a>
          </li>
          <li>
            <a href="http://www.w3.org/2004/10/xml.xsd">
              http://www.w3.org/2004/10/xml.xsd
            </a>
          </li>
          <li>
            <a href="http://www.w3.org/2001/03/xml.xsd">
              http://www.w3.org/2001/03/xml.xsd
            </a>
          </li>
        </ul>
      </div>
    </div>
  </xs:documentation>

```

```
</xs:annotation>  
</xs:schema>
```

Schema "XMLSchema.xsd"

Schema Source:

<http://www.w3.org/2001/XMLSchema.xsd>; see [XML source](#) [396]

Target Namespace:

<http://www.w3.org/2001/XMLSchema>

Version:

1.0

Defined Components:

[elements](#) ([top-level](#) / [other](#); 41 global + 28 local), [complexType](#)s (35), [simpleType](#)s (55), [element groups](#) (12), [attribute groups](#) (2)

Default Namespace-Qualified Form:

Local Elements: qualified; Local Attributes: unqualified

Default Block Attribute:

"#all" (*blocks all substitutions of elements and their types both through substitution groups and xsi:type attribute in instance XML documents*)

Imports Schemas (1):

[xml.xsd](#)

Annotation

Annotation 1 [[src](#), p. 397]:

Part 1 version: Id: structures.xsd,v 1.2 2004/01/15 11:34:25 ht Exp

Part 2 version: Id: datatypes.xsd,v 1.3 2004/01/23 18:11:13 ht Exp

Annotation 2 [[src](#), p. 397]:

The schema corresponding to this document is normative, with respect to the syntactic constraints it expresses in the XML Schema language. The documentation (within <documentation> elements) below, is not normative, but rather highlights important aspects of the W3C Recommendation of which this is a part

See: <http://www.w3.org/TR/2004/PER-xmlschema-1-20040318/structures.html>

Annotation 3 [[src](#), p. 397]:

The simpleType element and all of its members are defined towards the end of this schema document

Annotation 4 [[src](#), p. 408]:

simple type for the value of the 'namespace' attr of 'any' and 'anyAttribute'

Annotation 5 [[src](#), p. 408]:

Value is

##any - - any non-conflicting WFXML/attribute at all

##other - - any non-conflicting WFXML/attribute from namespace other than targetNS

##local - - any unqualified non-conflicting WFXML/attribute

one or - - any non-conflicting WFXML/attribute from more URI the listed namespaces references (space separated)

##targetNamespace or ##local may appear in the above list, to refer to the targetNamespace of the enclosing schema or an absent targetNamespace respectively

Annotation 6 [src, p. 413]:

notations for use within XML Schema schemas

Annotation 7 [src, p. 413]:

First the built-in primitive datatypes. These definitions are for information only, the real built-in definitions are magic.

Annotation 8 [src, p. 413]:

For each built-in datatype in this schema (both primitive and derived) can be uniquely addressed via a URI constructed as follows:

- 1) the base URI is the URI of the XML Schema namespace
- 2) the fragment identifier is the name of the datatype

For example, to address the int datatype, the URI is:

<http://www.w3.org/2001/XMLSchema#int>

Additionally, each facet definition element can be uniquely addressed via a URI constructed as follows:

- 1) the base URI is the URI of the XML Schema namespace
- 2) the fragment identifier is the name of the facet

For example, to address the maxInclusive facet, the URI is:

<http://www.w3.org/2001/XMLSchema#maxInclusive>

Additionally, each facet usage in a built-in datatype definition can be uniquely addressed via a URI constructed as follows:


- 1) the base URI is the URI of the XML Schema namespace
- 2) the fragment identifier is the name of the datatype, followed by a period (".") followed by the name of the facet

For example, to address the usage of the maxInclusive facet in the definition of int, the URI is:














<http://www.w3.org/2001/XMLSchema#int.maxInclusive>












Annotation 9 [src, p. 419]:













Now the derived primitive types













Top-Level Element Summary (root element candidates)		Page
 xs:schema	Type: anonymous complexType (extension of xs:openAttrs) [8] Content: complex, 8 attributes, attr. wildcard, 12 elements Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [8] Includes: definitions of 8 attributes, 5 elements Used: never	7



Other Element Summary (global + local unified by type)		Page
 xs:all	Type: xs:all [150] Content: complex, 3 attributes, attr. wildcard, 2 elements Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [13] Used: at 3 locations	12
 xs:all (in xs:group)	Type: anonymous complexType (restriction of xs:all) [16] Content: complex, 1 attribute, attr. wildcard, 2 elements Block: "#all" (blocks all substitutions of this element or its type) Defined: locally within xs:namedGroup complexType [214]; see XML source [16] Includes: definition of attr. wildcard; 2 attr. prohibitions	15








 xs:annotation	<p>Type: anonymous complexType (extension of xs:openAttrs) [18] Content: complex, 1 attribute, attr. wildcard, 2 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [18] Includes: definitions of 1 attribute, 2 elements Used: at 28 locations</p>	17
 xs:any	<p>Type: anonymous complexType (extension of xs:wildcard) [21] Content: complex, 5 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [21] Used: at 2 locations</p>	20
 xs:anyAttribute	<p>Type: xs:wildcard [260] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [24] Used: at 1 location</p>	23
 xs:appinfo	<p>Type: anonymous complexType [412] Content: mixed (<i>allows character data</i>), 1 attribute, attr. wildcard, elem. wildcard Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [25] Includes: definitions of 1 attribute, attr. wildcard, elem. wildcard Used: at 1 location</p>	25
 xs:attribute	<p>Type: xs:topLevelAttribute [246] Content: complex, 5 attributes, attr. wildcard, 2 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [28] Used: at 1 location</p>	27
 xs:attribute (type xs:attribute)	<p>Type: xs:attribute [158] Content: complex, 8 attributes, attr. wildcard, 2 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:attrDecls group [340]; see XML source [30]</p>	29
 xs:attributeGroup	<p>Type: xs:namedAttributeGroup [209] Content: complex, 2 attributes, attr. wildcard, 4 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [33] Used: at 1 location</p>	32
 xs:attributeGroup (type xs:attributeGroupRef)	<p>Type: xs:attributeGroupRef [165] Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:attrDecls group [341]; see XML source [34]</p>	34
 xs:choice	<p>Type: xs:explicitGroup [179] Content: complex, 3 attributes, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [37] Used: at 4 locations</p>	36
 xs:choice (in xs:group)	<p>Type: xs:simpleExplicitGroup [233] Content: complex, 1 attribute, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:namedGroup complexType [214]; see XML source [39]</p>	39
 xs:complexContent	<p>Type: anonymous complexType (extension of xs:annotated) [42] Content: complex, 2 attributes, attr. wildcard, 3 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [42] Includes: definitions of 1 attribute, 2 elements Used: at 1 location</p>	41
 xs:complexType	<p>Type: xs:topLevelComplexType [249] Content: complex, 6 attributes, attr. wildcard, 10 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [45] Used: at 1 location</p>	44
 xs:complexType (type xs:localComplexType)	<p>Type: xs:localComplexType [197] Content: complex, 2 attributes, attr. wildcard, 10 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally at 4 locations</p>	48










 xs:documentation	<p>Type: anonymous complexType [412] Content: mixed (<i>allows character data</i>), 2 attributes, attr. wildcard, elem. wildcard Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [51] Includes: definitions of 2 attributes, attr. wildcard, elem. wildcard Used: at 1 location</p>	51
 xs:element	<p>Type: xs:topLevelElement [253] Content: complex, 10 attributes, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [54] Used: at 1 location</p>	53
 xs:element (type xs:localElement)	<p>Type: xs:localElement [201] Content: complex, 11 attributes, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally at 2 locations</p>	57
 xs:element (type xs:narrowMaxMin)	<p>Type: xs:narrowMaxMin [215] Content: complex, 11 attributes, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:allModel group [339]; see XML source [62]</p>	61
 xs:enumeration	<p>Type: xs:noFixedFacet [220] Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [65] Used: at 1 location</p>	65
 xs:extension (in xs:complexContent)	<p>Type: xs:extensionType [183] Content: complex, 2 attributes, attr. wildcard, 8 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:complexContent element [43]; see XML source [68]</p>	67
 xs:extension (in xs:simpleContent)	<p>Type: xs:simpleExtensionType [236] Content: complex, 2 attributes, attr. wildcard, 4 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:simpleContent element [134]; see XML source [70]</p>	70
 xs:field	<p>Type: anonymous complexType (extension of xs:annotated) [73] Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [73] Includes: definition of 1 attribute Used: at 1 location</p>	72
 xs:fractionDigits	<p>Type: xs:numFacet [222] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [76] Used: at 1 location</p>	75
 xs:group	<p>Type: xs:namedGroup [212] Content: complex, 2 attributes, attr. wildcard, 4 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [78] Used: at 1 location</p>	77
 xs:group (type xs:groupRef)	<p>Type: xs:groupRef [192] Content: complex, 4 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally at 3 locations</p>	79
 xs:import	<p>Type: anonymous complexType (extension of xs:annotated) [82] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [82] Includes: definitions of 2 attributes Used: at 1 location</p>	81
 xs:include	<p>Type: anonymous complexType (extension of xs:annotated) [84] Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [84] Includes: definition of 1 attribute Used: at 1 location</p>	83











 xs:key	<p>Type: xs:keybase [195] Content: complex, 2 attributes, attr. wildcard, 3 elements Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [86] Used: at 1 location</p>	85
 xs:keyref	<p>Type: anonymous complexType (extension of xs:keybase) [88] Content: complex, 3 attributes, attr. wildcard, 3 elements Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [88] Includes: definition of 1 attribute Used: at 1 location</p>	87
 xs:length	<p>Type: xs:numFacet [222] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [91] Used: at 1 location</p>	90
 xs:list	<p>Type: anonymous complexType (extension of xs:annotated) [93] Content: complex, 2 attributes, attr. wildcard, 2 elements Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [93] Includes: definitions of 1 attribute, 1 element Used: at 1 location</p>	92
 xs:maxExclusive	<p>Type: xs:facet [186] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [94] Used: at 1 location</p>	94
 xs:maxInclusive	<p>Type: xs:facet [186] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [96] Used: at 1 location</p>	96
 xs:maxLength	<p>Type: xs:numFacet [222] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [99] Used: at 1 location</p>	98
 xs:minExclusive	<p>Type: xs:facet [186] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [100] Used: at 1 location</p>	100
 xs:minInclusive	<p>Type: xs:facet [186] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [102] Used: at 1 location</p>	102
 xs:minLength	<p>Type: xs:numFacet [222] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [105] Used: at 1 location</p>	104
 xs:notation	<p>Type: anonymous complexType (extension of xs:annotated) [107] Content: complex, 4 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [107] Includes: definitions of 3 attributes Used: at 1 location</p>	106
 xs:pattern	<p>Type: anonymous complexType (restriction of xs:noFixedFacet) [110] Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this element or its type) Defined: globally; see XML source [110] Includes: definitions of 1 attribute, attr. wildcard, 1 element Used: at 1 location</p>	109










 xs:redefine	<p>Type: anonymous complexType (extension of xs:openAttrs) [112] Content: complex, 2 attributes, attr. wildcard, 5 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [112] Includes: definitions of 2 attributes, 1 element Used: at 1 location</p>	<p>111</p>
 xs:restriction	<p>Type: anonymous complexType (extension of xs:annotated) [115] Content: complex, 2 attributes, attr. wildcard, 14 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [115] Includes: definition of 1 attribute Used: at 1 location</p>	<p>114</p>
 xs:restriction (in xs:complexContent)	<p>Type: xs:complexRestrictionType [167] Content: complex, 2 attributes, attr. wildcard, 8 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:complexContent element [43]; see XML source [119]</p>	<p>118</p>
 xs:restriction (in xs:simpleContent)	<p>Type: xs:simpleRestrictionType [239] Content: complex, 2 attributes, attr. wildcard, 17 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:simpleContent element [134]; see XML source [122]</p>	<p>121</p>
 xs:selector	<p>Type: anonymous complexType (extension of xs:annotated) [126] Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [126] Includes: definition of 1 attribute Used: at 1 location</p>	<p>125</p>
 xs:sequence	<p>Type: xs:explicitGroup [179] Content: complex, 3 attributes, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [129] Used: at 4 locations</p>	<p>128</p>
 xs:sequence (in xs:group)	<p>Type: xs:simpleExplicitGroup [233] Content: complex, 1 attribute, attr. wildcard, 6 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally within xs:namedGroup complexType [214]; see XML source [131]</p>	<p>131</p>
 xs:simpleContent	<p>Type: anonymous complexType (extension of xs:annotated) [134] Content: complex, 1 attribute, attr. wildcard, 3 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [134] Includes: definitions of 2 elements Used: at 1 location</p>	<p>133</p>
 xs:simpleType	<p>Type: xs:topLevelSimpleType [257] Content: complex, 3 attributes, attr. wildcard, 4 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [136] Used: at 1 location</p>	<p>135</p>
 xs:simpleType (type xs:localSimpleType)	<p>Type: xs:localSimpleType [206] Content: complex, 1 attribute, attr. wildcard, 4 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: locally at 9 locations</p>	<p>138</p>
 xs:totalDigits	<p>Type: anonymous complexType (restriction of xs:numFacet) [141] Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [141] Includes: definitions of 1 attribute, attr. wildcard, 1 element Used: at 1 location</p>	<p>140</p>
 xs:union	<p>Type: anonymous complexType (extension of xs:annotated) [143] Content: complex, 2 attributes, attr. wildcard, 2 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [143] Includes: definitions of 1 attribute, 1 element Used: at 1 location</p>	<p>142</p>



 xs:unique	Type: xs:keybase [195] Content: complex, 2 attributes , attr. wildcard , 3 elements Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [146] Used: at 1 location	145
 xs:whiteSpace	Type: anonymous complexType (restriction of xs:facet) [148] Content: complex, 3 attributes , attr. wildcard , 1 element Block: "#all" (<i>blocks all substitutions of this element or its type</i>) Defined: globally; see XML source [148] Includes: definitions of 1 attribute , attr. wildcard , 1 element Used: at 1 location	147

Complex Type Summary		Page
 xs:all	Only elements allowed inside Content: complex, 3 attributes , attr. wildcard , 2 elements Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [151] Includes: definitions of 2 attributes , attr. wildcard Used: at 2 locations	150
 xs:annotated	This type is extended by all types which allow annotation other than <schema> itself Content: complex, 1 attribute , attr. wildcard , 1 element Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [154] Includes: definitions of 1 attribute , 1 element Used: at 21 locations	153
 xs:anyType	Not the real urType, but as close an approximation as we can get in the XML representation Content: mixed (<i>allows character data</i>), attr. wildcard , elem. wildcard Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [157] Includes: definitions of attr. wildcard , elem. wildcard Used: at 1 location	156
 xs:attribute	Content: complex, 8 attributes , attr. wildcard , 2 elements Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [159] Includes: definitions of 5 attributes , 1 element Used: at 2 locations	158
 xs:attributeGroup	Content: complex, 3 attributes , attr. wildcard , 4 elements Abstract: (<i>cannot be assigned directly to elements used in instance XML documents</i>) Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [163] Used: at 2 locations	162
 xs:attributeGroupRef	Content: complex, 2 attributes , attr. wildcard , 1 element Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [166] Includes: definitions of 1 attribute , attr. wildcard , 1 element ; 1 attr. prohibition Used: at 1 location	165
 xs:complexRestrictionType	Content: complex, 2 attributes , attr. wildcard , 8 elements Block: "#all" (<i>blocks all substitutions of this complex type through xsi:type attribute in instance XML documents</i>) Defined: globally; see XML source [168] Includes: definitions of attr. wildcard , 1 element Used: at 1 location	167








 xs:complexType	<p>Content: complex, 6 attributes, attr. wildcard, 10 elements Abstract: <i>(cannot be assigned directly to elements used in instance XML documents)</i> Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [171] Includes: definitions of 5 attributes Used: at 2 locations</p>	<p>170</p>
 xs:element	<p>The element element can be used either at the top level to define an element-type binding globally, or within a content model to either reference a globally-defined element or type or declare an element-type binding locally.</p> <p>Content: complex, 14 attributes, attr. wildcard, 6 elements Abstract: <i>(cannot be assigned directly to elements used in instance XML documents)</i> Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [176] Includes: definitions of 9 attributes, 2 elements Used: at 2 locations</p>	<p>174</p>
 xs:explicitGroup	<p>group type for the three kinds of group</p> <p>Content: complex, 3 attributes, attr. wildcard, 6 elements Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [180] Includes: definitions of attr. wildcard, 1 element; 2 attr. prohibitions Used: at 4 locations</p>	<p>179</p>
 xs:extensionType	<p>Content: complex, 2 attributes, attr. wildcard, 8 elements Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [184] Includes: definition of 1 attribute Used: at 2 locations</p>	<p>183</p>
 xs:facet	<p>Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [187] Includes: definitions of 2 attributes Used: at 7 locations</p>	<p>186</p>
 xs:group	<p>group type for explicit groups, named top-level groups and group references</p> <p>Content: complex, 5 attributes, attr. wildcard, 7 elements Abstract: <i>(cannot be assigned directly to elements used in instance XML documents)</i> Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [189] Used: at 2 locations</p>	<p>188</p>
 xs:groupRef	<p>Content: complex, 4 attributes, attr. wildcard, 1 element Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [193] Includes: definitions of 1 attribute, attr. wildcard, 1 element; 1 attr. prohibition Used: at 1 location</p>	<p>192</p>
 xs:keybase	<p>Content: complex, 2 attributes, attr. wildcard, 3 elements Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [196] Includes: definitions of 1 attribute, 2 elements Used: at 3 locations</p>	<p>195</p>
 xs:localComplexType	<p>Content: complex, 2 attributes, attr. wildcard, 10 elements Block: "#all" <i>(blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</i> Defined: globally; see XML source [198] Includes: definitions of attr. wildcard, 1 element; 4 attr. prohibitions Used: at 1 location</p>	<p>197</p>






 xs:localElement	<p>Content: complex, 11 attributes, attr. wildcard, 6 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [202] Includes: definitions of attr. wildcard, 3 elements; 3 attr. prohibitions Used: at 2 locations</p>	201
 xs:localSimpleType	<p>Content: complex, 1 attribute, attr. wildcard, 4 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [207] Includes: definitions of attr. wildcard, 1 element; 2 attr. prohibitions Used: at 1 location</p>	206
 xs:namedAttributeGroup	<p>Content: complex, 2 attributes, attr. wildcard, 4 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [210] Includes: definitions of 1 attribute, attr. wildcard, 1 element; 1 attr. prohibition Used: at 1 location</p>	209
 xs:namedGroup	<p>Content: complex, 2 attributes, attr. wildcard, 4 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [213] Includes: definitions of 1 attribute, attr. wildcard, 4 elements; 3 attr. prohibitions Used: at 1 location</p>	212
 xs:narrowMaxMin	<p>restricted max/min Content: complex, 11 attributes, attr. wildcard, 6 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [216] Includes: definitions of 2 attributes, attr. wildcard, 3 elements Used: at 1 location</p>	215
 xs:noFixedFacet	<p>Content: complex, 2 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [221] Includes: definitions of attr. wildcard, 1 element; 1 attr. prohibition Used: at 2 locations</p>	220
 xs:numFacet	<p>Content: complex, 3 attributes, attr. wildcard, 1 element Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [223] Includes: definitions of 1 attribute, attr. wildcard, 1 element Used: at 5 locations</p>	222
 xs:openAttrs	<p>This type is extended by almost all schema types to allow attributes from other namespaces to be added to user schemas. Content: empty, attr. wildcard Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [225] Includes: definition of attr. wildcard Used: at 4 locations</p>	224
 xs:realGroup	<p>Content: complex, 5 attributes, attr. wildcard, 4 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [227] Includes: definitions of attr. wildcard, 4 elements Used: at 2 locations</p>	226
 xs:restrictionType	<p>Content: complex, 2 attributes, attr. wildcard, 21 elements Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents) Defined: globally; see XML source [230] Includes: definition of 1 attribute Used: at 2 locations</p>	229

 xs:simpleExplicitGroup	<p>Content: complex, 1 attribute, attr. wildcard, 6 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [234]</p> <p>Includes: definitions of attr. wildcard, 1 element; 2 attr. prohibitions</p> <p>Used: at 2 locations</p>	233
 xs:simpleExtensionType	<p>Content: complex, 2 attributes, attr. wildcard, 4 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [237]</p> <p>Includes: definitions of attr. wildcard, 1 element</p> <p>Used: at 1 location</p>	236
 xs:simpleRestrictionType	<p>Content: complex, 2 attributes, attr. wildcard, 17 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [240]</p> <p>Includes: definitions of attr. wildcard, 1 element</p> <p>Used: at 1 location</p>	239
 xs:simpleType	<p>Content: complex, 3 attributes, attr. wildcard, 4 elements</p> <p>Abstract: (cannot be assigned directly to elements used in instance XML documents)</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [244]</p> <p>Includes: definitions of 2 attributes</p> <p>Used: at 2 locations</p>	243
 xs:topLevelAttribute	<p>Content: complex, 5 attributes, attr. wildcard, 2 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [247]</p> <p>Includes: definitions of 1 attribute, attr. wildcard, 2 elements; 3 attr. prohibitions</p> <p>Used: at 1 location</p>	246
 xs:topLevelComplexType	<p>Content: complex, 6 attributes, attr. wildcard, 10 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [250]</p> <p>Includes: definitions of 1 attribute, attr. wildcard, 1 element</p> <p>Used: at 1 location</p>	249
 xs:topLevelElement	<p>Content: complex, 10 attributes, attr. wildcard, 6 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [254]</p> <p>Includes: definitions of 1 attribute, attr. wildcard, 3 elements; 4 attr. prohibitions</p> <p>Used: at 1 location</p>	253
 xs:topLevelSimpleType	<p>Content: complex, 3 attributes, attr. wildcard, 4 elements</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [258]</p> <p>Includes: definitions of 1 attribute, attr. wildcard, 1 element</p> <p>Used: at 1 location</p>	257
 xs:wildcard	<p>Content: complex, 3 attributes, attr. wildcard, 1 element</p> <p>Block: "#all" (blocks all substitutions of this complex type through xsi:type attribute in instance XML documents)</p> <p>Defined: globally; see XML source [261]</p> <p>Includes: definitions of 2 attributes</p> <p>Used: at 2 locations</p>	260



Simple Type Summary		Page
 xs:anyINNI	<p>for maxOccurs</p> <p>Defined: globally; see XML source [263]</p> <p>Used: at 3 locations</p>	263
 xs:anyURI	<p>Defined: globally; see XML source [265]</p> <p>Used: at 10 locations</p>	265
 xs:base64Binary	<p>Defined: globally; see XML source [267]</p> <p>Used: never</p>	267



 xs:blockSet	A utility type, not for public use #all or (possibly empty) subset of {substitution, extension, restriction} Defined: globally; see XML source [268] Used: at 2 locations	268
 xs:boolean	Defined: globally; see XML source [270] Used: at 6 locations	270
 xs:byte	Defined: globally; see XML source [272] Used: never	272
 xs:date	Defined: globally; see XML source [273] Used: never	273
 xs:dateTime	Defined: globally; see XML source [274] Used: never	274
 xs:decimal	Defined: globally; see XML source [275] Used: at 1 location	275
 xs:derivationControl	A utility type, not for public use Defined: globally; see XML source [277] Used: at 4 locations	277
 xs:derivationSet	A utility type, not for public use #all or (possibly empty) subset of {extension, restriction} Defined: globally; see XML source [279] Used: at 3 locations	279
 xs:double	Defined: globally; see XML source [281] Used: never	281
 xs:duration	Defined: globally; see XML source [282] Used: never	282
 xs:ENTITIES	Defined: globally; see XML source [283] Used: never	283
 xs:ENTITY	Defined: globally; see XML source [284] Used: at 1 location	284
 xs:float	Defined: globally; see XML source [285] Used: never	285
 xs:formChoice	A utility type, not for public use Defined: globally; see XML source [286] Used: at 4 locations	286
 xs:fullDerivationSet	A utility type, not for public use #all or (possibly empty) subset of {extension, restriction, list, union} Defined: globally; see XML source [287] Used: at 1 location	287
 xs:gDay	Defined: globally; see XML source [289] Used: never	289
 xs:gMonth	Defined: globally; see XML source [290] Used: never	290
 xs:gMonthDay	Defined: globally; see XML source [291] Used: never	291
 xs:gYear	Defined: globally; see XML source [292] Used: never	292
 xs:gYearMonth	Defined: globally; see XML source [293] Used: never	293
 xs:hexBinary	Defined: globally; see XML source [294] Used: never	294
 xs:ID	Defined: globally; see XML source [295] Used: at 5 locations	295
 xs:IDREF	Defined: globally; see XML source [296] Used: at 1 location	296
 xs:IDREFS	Defined: globally; see XML source [297] Used: never	297

 xs:int	Defined: globally; see XML source [298] Used: at 1 location	298
 xs:integer	Defined: globally; see XML source [299] Used: at 3 locations	299
 xs:language	Defined: globally; see XML source [301] Used: at 1 location	301
 xs:long	Defined: globally; see XML source [303] Used: at 1 location	303
 xs:Name	Defined: globally; see XML source [305] Used: at 1 location	304
 xs:namespaceList	A utility type, not for public use Defined: globally; see XML source [306] Used: at 1 location	306
 xs:NCName	Defined: globally; see XML source [309] Used: at 15 locations	308
 xs:negativeInteger	Defined: globally; see XML source [310] Used: never	310
 xs:NMTOKEN	Defined: globally; see XML source [312] Used: at 7 locations	311
 xs:NMTOKENS	Defined: globally; see XML source [313] Used: never	313
 xs:nonNegativeInteger	Defined: globally; see XML source [315] Used: at 7 locations	314
 xs:nonPositiveInteger	Defined: globally; see XML source [316] Used: at 1 location	316
 xs:normalizedString	Defined: globally; see XML source [318] Used: at 1 location	317
 xs:NOTATION	NOTATION cannot be used directly in a schema; rather a type must be derived from it by specifying at least one enumeration facet whose value is the name of a NOTATION declared in the schema. Defined: globally; see XML source [319] Used: never	319
 xs:positiveInteger	Defined: globally; see XML source [320] Used: at 1 location	320
 xs:public	A utility type, not for public use A public identifier, per ISO 8879 Defined: globally; see XML source [321] Used: at 1 location	321
 xs:QName	Defined: globally; see XML source [322] Used: at 12 locations	322
 xs:reducedDerivationControl	A utility type, not for public use Defined: globally; see XML source [324] Used: at 1 location	324
 xs:short	Defined: globally; see XML source [325] Used: at 1 location	325
 xs:simpleDerivationSet	#all or (possibly empty) subset of {restriction, union, list} A utility type, not for public use Defined: globally; see XML source [326] Used: at 1 location	326
 xs:string	Defined: globally; see XML source [329] Used: at 7 locations	328
 xs:time	Defined: globally; see XML source [330] Used: never	330
 xs:token	Defined: globally; see XML source [332] Used: at 13 locations	331

 xs:typeDerivationControl	A utility type, not for public use Defined: globally; see XML source [333] Used: at 1 location	333
 xs:unsignedByte	Defined: globally; see XML source [334] Used: never	334
 xs:unsignedInt	Defined: globally; see XML source [335] Used: at 1 location	335
 xs:unsignedLong	Defined: globally; see XML source [336] Used: at 1 location	336
 xs:unsignedShort	Defined: globally; see XML source [337] Used: at 1 location	337

Element Group Summary		Page
 xs:allModel	Content: 2 elements Defined: globally; see XML source [338] Includes: definitions of 2 elements Used: at 2 locations	338
 xs:attrDecls	Content: 3 elements Defined: globally; see XML source [340] Includes: definitions of 3 elements Used: at 8 locations	340
 xs:complexTypeModel	Content: 9 elements Defined: globally; see XML source [342] Includes: definitions of 2 elements Used: at 3 locations	342
 xs:facets	We should use a substitution group for facets, but that's ruled out because it would allow users to add their own, which we're not ready for yet. Content: 12 elements Defined: globally; see XML source [345] Includes: definitions of 12 elements Used: at 1 location	344
 xs:identityConstraint	The three kinds of identity constraints, all with type of or derived from 'keybase'. Content: 3 elements Defined: globally; see XML source [347] Includes: definitions of 3 elements Used: at 4 locations	347
 xs:nestedParticle	Content: 5 elements Defined: globally; see XML source [349] Includes: definitions of 5 elements Used: at 2 locations	349
 xs:particle	Content: 6 elements Defined: globally; see XML source [351] Includes: definitions of 6 elements Used: at 1 location	351
 xs:redefinable	This group is for the elements which can self-redefine (see <redefine> below). Content: 4 elements Defined: globally; see XML source [353] Includes: definitions of 4 elements Used: at 2 locations	353
 xs:schemaTop	This group is for the elements which occur freely at the top level of schemas. Content: 7 elements Defined: globally; see XML source [355] Includes: definitions of 3 elements Used: at 1 location	355
 xs:simpleDerivation	Content: 3 elements Defined: globally; see XML source [357] Includes: definitions of 3 elements Used: at 3 locations	357

 xs:simpleRestrictionModel	Content: 13 elements Defined: globally; see XML source [359] Includes: definition of 1 element Used: at 3 locations	358
 xs:typeDefParticle	'complexType' uses this Content: 4 elements Defined: globally; see XML source [361] Includes: definitions of 4 elements Used: at 4 locations	361

Attribute Group Summary		Page
 xs:defRef	for element, group and attributeGroup, which both define and reference Content: 2 attributes Defined: globally; see XML source [363] Includes: definitions of 2 attributes Used: at 4 locations	363
 xs:occurs	for all particles Content: 2 attributes Defined: globally; see XML source [365] Includes: definitions of 2 attributes Used: at 3 locations	365

Schema XML Source

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Schema schema for XML Schemas: Part 1: Structures -->
<!-- Note this schema is NOT the normative structures schema. -->
<!-- The prose copy in the structures REC is the normative -->
<!-- version (which shouldn't differ from this one except for -->
<!-- this comment and entity expansions, but just in case -->
<!DOCTYPE xs:schema PUBLIC "-//W3C//DTD XMLSCHEMA 200102//EN" "XMLSchema.dtd" [
  <!-- provide ID type information even for parsers which only read the
  internal subset -->
  <ATTLIST xs:schema id ID #IMPLIED>
  <ATTLIST xs:complexType id ID #IMPLIED>
  <ATTLIST xs:complexContent id ID #IMPLIED>
  <ATTLIST xs:simpleContent id ID #IMPLIED>
  <ATTLIST xs:extension id ID #IMPLIED>
  <ATTLIST xs:element id ID #IMPLIED>
  <ATTLIST xs:group id ID #IMPLIED>
  <ATTLIST xs:all id ID #IMPLIED>
  <ATTLIST xs:choice id ID #IMPLIED>
  <ATTLIST xs:sequence id ID #IMPLIED>
  <ATTLIST xs:any id ID #IMPLIED>
  <ATTLIST xs:anyAttribute id ID #IMPLIED>
  <ATTLIST xs:attribute id ID #IMPLIED>
  <ATTLIST xs:attributeGroup id ID #IMPLIED>
  <ATTLIST xs:unique id ID #IMPLIED>
  <ATTLIST xs:key id ID #IMPLIED>
  <ATTLIST xs:keyref id ID #IMPLIED>
  <ATTLIST xs:selector id ID #IMPLIED>
  <ATTLIST xs:field id ID #IMPLIED>
  <ATTLIST xs:include id ID #IMPLIED>
  <ATTLIST xs:import id ID #IMPLIED>
  <ATTLIST xs:redefine id ID #IMPLIED>
  <ATTLIST xs:notation id ID #IMPLIED>
  <!--
  keep this schema XML1.0 DTD valid
  -->
  <ENTITY % schemaAttrs 'xmlns:hfp CDATA #IMPLIED'>
  <ELEMENT hfp:hasFacet EMPTY>
  <ATTLIST hfp:hasFacet name NMTOKEN #REQUIRED>
  <ELEMENT hfp:hasProperty EMPTY>
  <ATTLIST hfp:hasProperty name NMTOKEN #REQUIRED>
  <ATTLIST hfp:hasProperty value CDATA #REQUIRED>
  <!--
  Make sure that processors that do not read the external
  subset will know about the various IDs we declare
  -->
  ]
  
```



```

<!ATTLIST xs:simpleType id ID #IMPLIED>
<!ATTLIST xs:maxExclusive id ID #IMPLIED>
<!ATTLIST xs:minExclusive id ID #IMPLIED>
<!ATTLIST xs:maxInclusive id ID #IMPLIED>
<!ATTLIST xs:minInclusive id ID #IMPLIED>
<!ATTLIST xs:totalDigits id ID #IMPLIED>
<!ATTLIST xs:fractionDigits id ID #IMPLIED>
<!ATTLIST xs:length id ID #IMPLIED>
<!ATTLIST xs:minLength id ID #IMPLIED>
<!ATTLIST xs:maxLength id ID #IMPLIED>
<!ATTLIST xs:enumeration id ID #IMPLIED>
<!ATTLIST xs:pattern id ID #IMPLIED>
<!ATTLIST xs:appinfo id ID #IMPLIED>
<!ATTLIST xs:documentation id ID #IMPLIED>
<!ATTLIST xs:list id ID #IMPLIED>
<!ATTLIST xs:union id ID #IMPLIED>
]>
<xs:schema blockDefault="#all" elementFormDefault="qualified" targetNamespace="http://www.w3.org/2001/XMLSchema"
version="1.0" xml:lang="EN" xmlns:hfp="http://www.w3.org/2001/XMLSchema-hasFacetAndProperty"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation>
      Part 1 version: Id: structures.xsd,v 1.2 2004/01/15 11:34:25 ht Exp
      Part 2 version: Id: datatypes.xsd,v 1.3 2004/01/23 18:11:13 ht Exp
    </xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/2004/PER-xmlschema-1-20040318/structures.html">
      The schema corresponding to this document is normative,
      with respect to the syntactic constraints it expresses in the
      XML Schema language. The documentation (within &lt;documentation&gt; elements)
      below, is not normative, but rather highlights important aspects of
      the W3C Recommendation of which this is a part
    </xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation>
      The simpleType element and all of its members are defined
      towards the end of this schema document
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd">
    <xs:annotation>
      <xs:documentation>
        Get access to the xml: attribute groups for xml:lang
        as declared on 'schema' and 'documentation' below
      </xs:documentation>
    </xs:annotation>
  </xs:import>
  <xs:complexType name="openAttrs">
    <xs:annotation>
      <xs:documentation>
        This type is extended by almost all schema types
        to allow attributes from other namespaces to be
        added to user schemas.
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="annotated">
    <xs:annotation>
      <xs:documentation>
        This type is extended by all types which allow annotation
        other than &lt;schema&gt; itself
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:group name="schemaTop">
  <xs:annotation>
    <xs:documentation>
      This group is for the
      elements which occur freely at the top level of schemas.
      All of their types are based on the "annotated" type by extension.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:group ref="xs:redefinable"/>
    <xs:element ref="xs:element"/>
    <xs:element ref="xs:attribute"/>
    <xs:element ref="xs:notation"/>
  </xs:choice>
</xs:group>
<xs:group name="redefinable">
  <xs:annotation>
    <xs:documentation>
      This group is for the
      elements which can self-redefine (see &lt;redefine&gt; below).
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element ref="xs:simpleType"/>
    <xs:element ref="xs:complexType"/>
    <xs:element ref="xs:group"/>
    <xs:element ref="xs:attributeGroup"/>
  </xs:choice>
</xs:group>
<xs:simpleType name="formChoice">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="qualified"/>
    <xs:enumeration value="unqualified"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="reducedDerivationControl">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:derivationControl">
    <xs:enumeration value="extension"/>
    <xs:enumeration value="restriction"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="derivationSet">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
    <xs:documentation>
      #all or (possibly empty) subset of {extension, restriction}
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="#all"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:list itemType="xs:reducedDerivationControl"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:simpleType name="typeDerivationControl">

```

```

<xs:annotation>
  <xs:documentation>
    A utility type, not for public use
  </xs:documentation>
</xs:annotation>
<xs:restriction base="xs:derivationControl">
  <xs:enumeration value="extension"/>
  <xs:enumeration value="restriction"/>
  <xs:enumeration value="list"/>
  <xs:enumeration value="union"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="fullDerivationSet">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
    <xs:documentation>
      #all or (possibly empty) subset of {extension, restriction, list, union}
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="#all"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:list itemType="xs:typeDerivationControl"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:element id="schema" name="schema">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-schema"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:sequence>
          <xs:choice maxOccurs="unbounded" minOccurs="0">
            <xs:element ref="xs:include"/>
            <xs:element ref="xs:import"/>
            <xs:element ref="xs:redefine"/>
            <xs:element ref="xs:annotation"/>
          </xs:choice>
          <xs:sequence maxOccurs="unbounded" minOccurs="0">
            <xs:group ref="xs:schemaTop">
              <xs:element maxOccurs="unbounded" minOccurs="0" ref="xs:annotation"/>
            </xs:group>
          </xs:sequence>
        </xs:sequence>
        <xs:attribute name="targetNamespace" type="xs:anyURI"/>
        <xs:attribute name="version" type="xs:token"/>
        <xs:attribute default="" name="finalDefault" type="xs:fullDerivationSet" use="optional"/>
        <xs:attribute default="" name="blockDefault" type="xs:blockSet" use="optional"/>
        <xs:attribute default="unqualified" name="attributeFormDefault" type="xs:formChoice" use="optional"/>
        <xs:attribute default="unqualified" name="elementFormDefault" type="xs:formChoice" use="optional"/>
        <xs:attribute name="id" type="xs:ID"/>
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:key name="element">
    <xs:selector xpath="xs:element"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="attribute">
    <xs:selector xpath="xs:attribute"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="type">
    <xs:selector xpath="xs:complexType|xs:simpleType"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="group">

```

```

    <xs:selector xpath="xs:group"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="attributeGroup">
    <xs:selector xpath="xs:attributeGroup"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="notation">
    <xs:selector xpath="xs:notation"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:key name="identityConstraint">
    <xs:selector xpath="./xs:key|./xs:unique|./xs:keyref"/>
    <xs:field xpath="@name"/>
  </xs:key>
</xs:element>
<xs:simpleType name="allNN1">
  <xs:annotation>
    <xs:documentation>
      for maxOccurs
    </xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="xs:nonNegativeInteger">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="unbounded"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:attributeGroup name="occurs">
  <xs:annotation>
    <xs:documentation>
      for all particles
    </xs:documentation>
  </xs:annotation>
  <xs:attribute default="1" name="minOccurs" type="xs:nonNegativeInteger" use="optional"/>
  <xs:attribute default="1" name="maxOccurs" type="xs:allNN1" use="optional"/>
</xs:attributeGroup>
<xs:attributeGroup name="defRef">
  <xs:annotation>
    <xs:documentation>
      for element, group and attributeGroup,
      which both define and reference
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="name" type="xs:NCName"/>
  <xs:attribute name="ref" type="xs:QName"/>
</xs:attributeGroup>
<xs:group name="typeDefParticle">
  <xs:annotation>
    <xs:documentation>
      'complexType' uses this
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element name="group" type="xs:groupRef"/>
    <xs:element ref="xs:all"/>
    <xs:element ref="xs:choice"/>
    <xs:element ref="xs:sequence"/>
  </xs:choice>
</xs:group>
<xs:group name="nestedParticle">
  <xs:choice>
    <xs:element name="element" type="xs:localElement"/>
    <xs:element name="group" type="xs:groupRef"/>
    <xs:element ref="xs:choice"/>
    <xs:element ref="xs:sequence"/>
    <xs:element ref="xs:any"/>
  </xs:choice>
</xs:group>
<xs:group name="particle">
  <xs:choice>
    <xs:element name="element" type="xs:localElement"/>
    <xs:element name="group" type="xs:groupRef"/>
  </xs:choice>
</xs:group>

```

```

<xs:element ref="xs:all"/>
<xs:element ref="xs:choice"/>
<xs:element ref="xs:sequence"/>
<xs:element ref="xs:any"/>
</xs:choice>
</xs:group>
<xs:complexType name="attribute">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
      </xs:sequence>
      <xs:attributeGroup ref="xs:defRef"/>
      <xs:attribute name="type" type="xs:QName"/>
      <xs:attribute default="optional" name="use" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="prohibited"/>
            <xs:enumeration value="optional"/>
            <xs:enumeration value="required"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="default" type="xs:string"/>
      <xs:attribute name="fixed" type="xs:string"/>
      <xs:attribute name="form" type="xs:formChoice"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="topLevelAttribute">
  <xs:complexContent>
    <xs:restriction base="xs:attribute">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
      </xs:sequence>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:attribute name="form" use="prohibited"/>
      <xs:attribute name="use" use="prohibited"/>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:group name="attrDecls">
  <xs:sequence>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:element name="attribute" type="xs:attribute"/>
      <xs:element name="attributeGroup" type="xs:attributeGroupRef"/>
    </xs:choice>
    <xs:element minOccurs="0" ref="xs:anyAttribute"/>
  </xs:sequence>
</xs:group>
<xs:element id="anyAttribute" name="anyAttribute" type="xs:wildcard">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-anyAttribute"/>
  </xs:annotation>
</xs:element>
<xs:group name="complexTypeModel">
  <xs:choice>
    <xs:element ref="xs:simpleContent"/>
    <xs:element ref="xs:complexContent"/>
  </xs:choice>
  <xs:sequence>
    <xs:annotation>
      <xs:documentation>
        This branch is short for
        &lt;complexContent&gt;
        &lt;restriction base="xs:anyType"&gt;
        ...
        &lt;/restriction&gt;
        &lt;/complexContent&gt;
      </xs:documentation>
    </xs:annotation>
    <xs:group minOccurs="0" ref="xs:typeDefParticle"/>
    <xs:group ref="xs:attrDecls"/>
  </xs:sequence>
</xs:group>

```

```

</xs:sequence>
</xs:choice>
</xs:group>
<xs:complexType abstract="true" name="complexType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group ref="xs:complexTypeModel"/>
      <xs:attribute name="name" type="xs:NCName">
        <xs:annotation>
          <xs:documentation>
            Will be restricted to required or forbidden
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute default="false" name="mixed" type="xs:boolean" use="optional">
        <xs:annotation>
          <xs:documentation>
            Not allowed if simpleContent child is chosen.
            May be overridden by setting on complexContent child.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute default="false" name="abstract" type="xs:boolean" use="optional"/>
      <xs:attribute name="final" type="xs:derivationSet"/>
      <xs:attribute name="block" type="xs:derivationSet"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="topLevelComplexType">
  <xs:complexContent>
    <xs:restriction base="xs:complexType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:complexTypeModel"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="localComplexType">
  <xs:complexContent>
    <xs:restriction base="xs:complexType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:complexTypeModel"/>
      </xs:sequence>
      <xs:attribute name="name" use="prohibited"/>
      <xs:attribute name="abstract" use="prohibited"/>
      <xs:attribute name="final" use="prohibited"/>
      <xs:attribute name="block" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="restrictionType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:group ref="xs:typeDefParticle"/>
          <xs:group ref="xs:simpleRestrictionModel"/>
        </xs:choice>
        <xs:group ref="xs:attrDecls"/>
      </xs:sequence>
      <xs:attribute name="base" type="xs:QName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="complexRestrictionType">
  <xs:complexContent>
    <xs:restriction base="xs:restrictionType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">

```

```

    <xs:annotation>
      <xs:documentation>
        This choice is added simply to
        make this a valid restriction per the REC
      </xs:documentation>
    </xs:annotation>
    <xs:group ref="xs:typeDefParticle"/>
  </xs:choice>
  <xs:group ref="xs:attrDecls"/>
</xs:sequence>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="extensionType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:group minOccurs="0" ref="xs:typeDefParticle"/>
        <xs:group ref="xs:attrDecls"/>
      </xs:sequence>
      <xs:attribute name="base" type="xs:QName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element id="complexContent" name="complexContent">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-complexContent"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:choice>
          <xs:element name="restriction" type="xs:complexRestrictionType"/>
          <xs:element name="extension" type="xs:extensionType"/>
        </xs:choice>
        <xs:attribute name="mixed" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>
              Overrides any setting on complexType parent.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:complexType name="simpleRestrictionType">
  <xs:complexContent>
    <xs:restriction base="xs:restrictionType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              This choice is added simply to
              make this a valid restriction per the REC
            </xs:documentation>
          </xs:annotation>
          <xs:group ref="xs:simpleRestrictionModel"/>
        </xs:choice>
        <xs:group ref="xs:attrDecls"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="simpleExtensionType">
  <xs:complexContent>
    <xs:restriction base="xs:extensionType">
      <xs:sequence>
        <xs:annotation>
          <xs:documentation>
            No typeDefParticle group reference
          </xs:documentation>
        </xs:annotation>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:annotation>
    <xs:element minOccurs="0" ref="xs:annotation"/>
    <xs:group ref="xs:attrDecls"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:element id="simpleContent" name="simpleContent">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-simpleContent"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:choice>
          <xs:element name="restriction" type="xs:simpleRestrictionType"/>
          <xs:element name="extension" type="xs:simpleExtensionType"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="complexType" name="complexType" type="xs:topLevelComplexType">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-complexType"/>
  </xs:annotation>
</xs:element>
<xs:simpleType name="blockSet">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
    <xs:documentation>
      #all or (possibly empty) subset of {substitution, extension, restriction}
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="#all"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:list>
        <xs:simpleType>
          <xs:restriction base="xs:derivationControl">
            <xs:enumeration value="extension"/>
            <xs:enumeration value="restriction"/>
            <xs:enumeration value="substitution"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:list>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:complexType abstract="true" name="element">
  <xs:annotation>
    <xs:documentation>
      The element element can be used either at the top level to define an element-type binding globally, or within a content model to either reference a globally-defined element or type or declare an element-type binding locally. The ref form is not allowed at the top level.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```



```

</xs:sequence>
<xs:attributeGroup ref="xs:defRef"/>
<xs:attribute name="type" type="xs:QName"/>
<xs:attribute name="substitutionGroup" type="xs:QName"/>
<xs:attributeGroup ref="xs:occurs"/>
<xs:attribute name="default" type="xs:string"/>
<xs:attribute name="fixed" type="xs:string"/>
<xs:attribute default="false" name="nillable" type="xs:boolean" use="optional"/>
<xs:attribute default="false" name="abstract" type="xs:boolean" use="optional"/>
<xs:attribute name="final" type="xs:derivationSet"/>
<xs:attribute name="block" type="xs:blockSet"/>
<xs:attribute name="form" type="xs:formChoice"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="topLevelElement">
  <xs:complexContent>
    <xs:restriction base="xs:element">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:attribute name="form" use="prohibited"/>
      <xs:attribute name="minOccurs" use="prohibited"/>
      <xs:attribute name="maxOccurs" use="prohibited"/>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="localElement">
  <xs:complexContent>
    <xs:restriction base="xs:element">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
      <xs:attribute name="substitutionGroup" use="prohibited"/>
      <xs:attribute name="final" use="prohibited"/>
      <xs:attribute name="abstract" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:element id="element" name="element" type="xs:topLevelElement">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-element"/>
  </xs:annotation>
</xs:element>
<xs:complexType abstract="true" name="group">
  <xs:annotation>
    <xs:documentation>
      group type for explicit groups, named top-level groups and group references
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:particle"/>
      <xs:attributeGroup ref="xs:defRef"/>
      <xs:attributeGroup ref="xs:occurs"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="realGroup">
  <xs:complexContent>

```

```

<xs:restriction base="xs:group">
  <xs:sequence>
    <xs:element minOccurs="0" ref="xs:annotation"/>
    <xs:choice maxOccurs="1" minOccurs="0">
      <xs:element ref="xs:all"/>
      <xs:element ref="xs:choice"/>
      <xs:element ref="xs:sequence"/>
    </xs:choice>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="namedGroup">
  <xs:complexContent>
    <xs:restriction base="xs:realGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice maxOccurs="1" minOccurs="1">
          <xs:element name="all">
            <xs:complexType>
              <xs:complexContent>
                <xs:restriction base="xs:all">
                  <xs:group ref="xs:allModel"/>
                  <xs:attribute name="minOccurs" use="prohibited"/>
                  <xs:attribute name="maxOccurs" use="prohibited"/>
                  <xs:anyAttribute namespace="##other" processContents="lax"/>
                </xs:restriction>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="choice" type="xs:simpleExplicitGroup"/>
          <xs:element name="sequence" type="xs:simpleExplicitGroup"/>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:attribute name="minOccurs" use="prohibited"/>
      <xs:attribute name="maxOccurs" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="groupRef">
  <xs:complexContent>
    <xs:restriction base="xs:realGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="ref" type="xs:QName" use="required"/>
      <xs:attribute name="name" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="explicitGroup">
  <xs:annotation>
    <xs:documentation>
      group type for the three kinds of group
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:restriction base="xs:group">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:nestedParticle"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="prohibited"/>
      <xs:attribute name="ref" type="xs:QName" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="simpleExplicitGroup">

```

```

<xs:complexContent>
  <xs:restriction base="xs:explicitGroup">
    <xs:sequence>
      <xs:element minOccurs="0" ref="xs:annotation"/>
      <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:nestedParticle"/>
    </xs:sequence>
    <xs:attribute name="minOccurs" use="prohibited"/>
    <xs:attribute name="maxOccurs" use="prohibited"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:group name="allModel">
  <xs:sequence>
    <xs:element minOccurs="0" ref="xs:annotation"/>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          This choice with min/max is here to
          avoid a pbIm with the Elt:All/Choice/Seq
          Particle derivation constraint
        </xs:documentation>
      </xs:annotation>
      <xs:element name="element" type="xs:narrowMaxMin"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
<xs:complexType name="narrowMaxMin">
  <xs:annotation>
    <xs:documentation>restricted max/min</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:restriction base="xs:localElement">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:choice minOccurs="0">
          <xs:element name="simpleType" type="xs:localSimpleType"/>
          <xs:element name="complexType" type="xs:localComplexType"/>
        </xs:choice>
        <xs:group maxOccurs="unbounded" minOccurs="0" ref="xs:identityConstraint"/>
      </xs:sequence>
      <xs:attribute default="1" name="minOccurs" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:nonNegativeInteger">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute default="1" name="maxOccurs" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:allNNI">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="all">
  <xs:annotation>
    <xs:documentation>
      Only elements allowed inside
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:restriction base="xs:explicitGroup">
      <xs:group ref="xs:allModel"/>
      <xs:attribute default="1" name="minOccurs" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:nonNegativeInteger">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute default="1" name="maxOccurs" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:allNNI">
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:element id="all" name="all" type="xs:all">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-all"/>
  </xs:annotation>
</xs:element>
<xs:element id="choice" name="choice" type="xs:explicitGroup">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-choice"/>
  </xs:annotation>
</xs:element>
<xs:element id="sequence" name="sequence" type="xs:explicitGroup">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-sequence"/>
  </xs:annotation>
</xs:element>
<xs:element id="group" name="group" type="xs:namedGroup">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-group"/>
  </xs:annotation>
</xs:element>
<xs:complexType name="wildcard">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:attribute default="##any" name="namespace" type="xs:namespaceList" use="optional"/>
      <xs:attribute default="strict" name="processContents" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="skip"/>
            <xs:enumeration value="lax"/>
            <xs:enumeration value="strict"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element id="any" name="any">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-any"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:wildcard">
        <xs:attributeGroup ref="xs:occurs"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:annotation>
  <xs:documentation>
    simple type for the value of the 'namespace' attr of
    'any' and 'anyAttribute'
  </xs:documentation>
</xs:annotation>
<xs:annotation>
  <xs:documentation>
Value is
##any - - any non-conflicting WFXML/attribute at all

##other - - any non-conflicting WFXML/attribute from
namespace other than targetNS

```

##local - - any unqualified non-conflicting WFXML/attribute

one or - - any non-conflicting WFXML/attribute from more URI the listed namespaces references (space separated)

##targetNamespace or **##local** may appear in the above list, to refer to the targetNamespace of the enclosing schema or an absent targetNamespace respectively

```

</xs:documentation>
</xs:annotation>
<xs:simpleType name="namespaceList">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="##any"/>
        <xs:enumeration value="##other"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:list>
        <xs:simpleType>
          <xs:union memberTypes="xs:anyURI">
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="##targetNamespace"/>
                <xs:enumeration value="##local"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
      </xs:list>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:element id="attribute" name="attribute" type="xs:topLevelAttribute">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-attribute"/>
  </xs:annotation>
</xs:element>
<xs:complexType abstract="true" name="attributeGroup">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group ref="xs:attrDecls" />
      <xs:attributeGroup ref="xs:defRef" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="namedAttributeGroup">
  <xs:complexContent>
    <xs:restriction base="xs:attributeGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation" />
        <xs:group ref="xs:attrDecls" />
      </xs:sequence>
      <xs:attribute name="name" type="xs:NCName" use="required"/>
      <xs:attribute name="ref" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="attributeGroupRef">
  <xs:complexContent>
    <xs:restriction base="xs:attributeGroup">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation" />
      </xs:sequence>
      <xs:attribute name="ref" type="xs:QName" use="required"/>
      <xs:attribute name="name" use="prohibited"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

```

    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:element id="attributeGroup" name="attributeGroup" type="xs:namedAttributeGroup">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-attributeGroup"/>
  </xs:annotation>
</xs:element>
<xs:element id="include" name="include">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-include"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="schemaLocation" type="xs:anyURI" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="redefine" name="redefine">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-redefine"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:choice maxOccurs="unbounded" minOccurs="0">
          <xs:element ref="xs:annotation"/>
          <xs:group ref="xs:redefinable"/>
        </xs:choice>
        <xs:attribute name="schemaLocation" type="xs:anyURI" use="required"/>
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="import" name="import">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-import"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="namespace" type="xs:anyURI"/>
        <xs:attribute name="schemaLocation" type="xs:anyURI"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="selector" name="selector">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-selector"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="xpath" use="required">
          <xs:simpleType>
            <xs:annotation>
              <xs:documentation>
                A subset of XPath expressions for use in selectors
              </xs:documentation>
              <xs:documentation>
                A utility type, not for public use
              </xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:token">
              <xs:annotation>
                <xs:documentation>
                  The following pattern is intended to allow XPath expressions per the following EBNF:
                </xs:documentation>
              </xs:annotation>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

```

Selector ::= Path ( '|' Path )*
Path ::= ('./')? Step ( '/' Step )*
Step ::= '.' | NameTest
NameTest ::= QName | '*' | NCName ':' '*'
child:: is also allowed
</xs:documentation>
</xs:annotation>
<xs:pattern
value="(\\./)?(((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)/((((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)*)|(\\./)?(((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)/((((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)*)*)"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element id="field" name="field">
<xs:annotation>
<xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-field"/>
</xs:annotation>
<xs:complexType>
<xs:complexContent>
<xs:extension base="xs:annotated">
<xs:attribute name="xpath" use="required">
<xs:simpleType>
<xs:annotation>
<xs:documentation>
A subset of XPath expressions for use in fields
</xs:documentation>
<xs:documentation>
A utility type, not for public use
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:token">
<xs:annotation>
<xs:documentation>
The following pattern is intended to allow XPath expressions per the same EBNF as for selector, with the following change:
Path ::= ('./')? ( Step '/' )* ( Step | '@' NameTest )
</xs:documentation>
</xs:annotation>
<xs:pattern
value="(\\./)?(((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)/*(((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)|((attribute::|@)((\\i\\c*?)?\\i\\c*|\\*)))(\\|\\./)?(((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)/*(((child::)?((\\i\\c*?)?\\i\\c*|\\*))|\\.)|((attribute::|@)((\\i\\c*?)?\\i\\c*|\\*)))*"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:complexType name="keybase">
<xs:complexContent>
<xs:extension base="xs:annotated">
<xs:sequence>
<xs:element ref="xs:selector"/>
<xs:element maxOccurs="unbounded" minOccurs="1" ref="xs:field"/>
</xs:sequence>
<xs:attribute name="name" type="xs:NCName" use="required"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:group name="identityConstraint">
<xs:annotation>
<xs:documentation>
The three kinds of identity constraints, all with type of or derived from 'keybase'.
</xs:documentation>
</xs:annotation>
</xs:choice>
<xs:element ref="xs:unique"/>

```



```

    <xs:element ref="xs:key"/>
    <xs:element ref="xs:keyref"/>
  </xs:choice>
</xs:group>
<xs:element id="unique" name="unique" type="xs:keybase">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-unique"/>
  </xs:annotation>
</xs:element>
<xs:element id="key" name="key" type="xs:keybase">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-key"/>
  </xs:annotation>
</xs:element>
<xs:element id="keyref" name="keyref">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-keyref"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:keybase">
        <xs:attribute name="refer" type="xs:QName" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="notation" name="notation">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-notation"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:attribute name="name" type="xs:NCName" use="required"/>
        <xs:attribute name="public" type="xs:public"/>
        <xs:attribute name="system" type="xs:anyURI"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="public">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
    <xs:documentation>
      A public identifier, per ISO 8879
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token"/>
</xs:simpleType>
<xs:element id="appinfo" name="appinfo">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-appinfo"/>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:sequence maxOccurs="unbounded" minOccurs="0">
      <xs:any processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="source" type="xs:anyURI"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:element>
<xs:element id="documentation" name="documentation">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-documentation"/>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:sequence maxOccurs="unbounded" minOccurs="0">
      <xs:any processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="source" type="xs:anyURI"/>
    <xs:attribute ref="xml:lang"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

```



```

</xs:element>
<xs:element id="annotation" name="annotation">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-1/#element-annotation"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:choice maxOccurs="unbounded" minOccurs="0">
          <xs:element ref="xs:appinfo"/>
          <xs:element ref="xs:documentation"/>
        </xs:choice>
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:annotation>
  <xs:documentation>
    notations for use within XML Schema schemas
  </xs:documentation>
</xs:annotation>
<xs:notation name="XMLSchemaStructures" public="structures"
system="http://www.w3.org/2000/08/XMLSchema.xsd"/>
<xs:notation name="XML" public="REC-xml-19980210" system="http://www.w3.org/TR/1998/REC-xml-19980210"/>
<xs:complexType mixed="true" name="anyType">
  <xs:annotation>
    <xs:documentation>
      Not the real urType, but as close an approximation as we can
      get in the XML representation
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any maxOccurs="unbounded" minOccurs="0" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:annotation>
  <xs:documentation>
    First the built-in primitive datatypes. These definitions are for
    information only, the real built-in definitions are magic.
  </xs:documentation>
  <xs:documentation>
    For each built-in datatype in this schema (both primitive and
    derived) can be uniquely addressed via a URI constructed
    as follows:
    1) the base URI is the URI of the XML Schema namespace
    2) the fragment identifier is the name of the datatype

    For example, to address the int datatype, the URI is:

    http://www.w3.org/2001/XMLSchema#int

    Additionally, each facet definition element can be uniquely
    addressed via a URI constructed as follows:
    1) the base URI is the URI of the XML Schema namespace
    2) the fragment identifier is the name of the facet

    For example, to address the maxInclusive facet, the URI is:

    http://www.w3.org/2001/XMLSchema#maxInclusive

    Additionally, each facet usage in a built-in datatype definition
    can be uniquely addressed via a URI constructed as follows:
    1) the base URI is the URI of the XML Schema namespace
    2) the fragment identifier is the name of the datatype, followed
    by a period (".") followed by the name of the facet

    For example, to address the usage of the maxInclusive facet in
    the definition of int, the URI is:

    http://www.w3.org/2001/XMLSchema#int.maxInclusive
  </xs:documentation>
</xs:annotation>
<xs:simpleType id="string" name="string">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
    </xs:appinfo>
  </xs:annotation>

```

```

    <hfp:hasFacet name="maxLength"/>
    <hfp:hasFacet name="pattern"/>
    <hfp:hasFacet name="enumeration"/>
    <hfp:hasFacet name="whiteSpace"/>
    <hfp:hasProperty name="ordered" value="false"/>
    <hfp:hasProperty name="bounded" value="false"/>
    <hfp:hasProperty name="cardinality" value="countably infinite"/>
    <hfp:hasProperty name="numeric" value="false"/>
  </xs:appinfo>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#string"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace id="string.preserve" value="preserve"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType id="boolean" name="boolean">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#boolean"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="boolean.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="float" name="float">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="total"/>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
      <hfp:hasProperty name="numeric" value="true"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#float"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="float.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="double" name="double">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="total"/>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
      <hfp:hasProperty name="numeric" value="true"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#double"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="double.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="decimal" name="decimal">

```

```

<xs:annotation>
  <xs:appinfo>
    <hfp:hasFacet name="totalDigits"/>
    <hfp:hasFacet name="fractionDigits"/>
    <hfp:hasFacet name="pattern"/>
    <hfp:hasFacet name="whiteSpace"/>
    <hfp:hasFacet name="enumeration"/>
    <hfp:hasFacet name="maxInclusive"/>
    <hfp:hasFacet name="maxExclusive"/>
    <hfp:hasFacet name="minInclusive"/>
    <hfp:hasFacet name="minExclusive"/>
    <hfp:hasProperty name="ordered" value="total"/>
    <hfp:hasProperty name="bounded" value="false"/>
    <hfp:hasProperty name="cardinality" value="countably infinite"/>
    <hfp:hasProperty name="numeric" value="true"/>
  </xs:appinfo>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#decimal"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="decimal.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType id="duration" name="duration">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#duration"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="duration.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="dateTime" name="dateTime">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#dateTime"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="dateTime.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="time" name="time">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
    </xs:appinfo>
  </xs:annotation>

```

```

    <hfp:hasProperty name="ordered" value="partial"/>
    <hfp:hasProperty name="bounded" value="false"/>
    <hfp:hasProperty name="cardinality" value="countably infinite"/>
    <hfp:hasProperty name="numeric" value="false"/>
  </xs:appinfo>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#time"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="time.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType id="date" name="date">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#date"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="date.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="gYearMonth" name="gYearMonth">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gYearMonth"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gYearMonth.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="gYear" name="gYear">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gYear"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gYear.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType id="gMonthDay" name="gMonthDay">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gMonthDay"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gMonthDay.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="gDay" name="gDay">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gDay"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gDay.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="gMonth" name="gMonth">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="maxInclusive"/>
      <hfp:hasFacet name="maxExclusive"/>
      <hfp:hasFacet name="minInclusive"/>
      <hfp:hasFacet name="minExclusive"/>
      <hfp:hasProperty name="ordered" value="partial"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#gMonth"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="gMonth.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="hexBinary" name="hexBinary">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
    </xs:appinfo>
  </xs:annotation>

```

```

    <hfp:hasProperty name="cardinality" value="countably infinite"/>
    <hfp:hasProperty name="numeric" value="false"/>
  </xs:appinfo>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#binary"/>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="hexBinary.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType id="base64Binary" name="base64Binary">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#base64Binary"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="base64Binary.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="anyURI" name="anyURI">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#anyURI"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="anyURI.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="QName" name="QName">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#QName"/>
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace fixed="true" id="QName.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="NOTATION" name="NOTATION">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
    </xs:appinfo>

```



```

    <hfp:hasFacet name="maxLength"/>
    <hfp:hasFacet name="pattern"/>
    <hfp:hasFacet name="enumeration"/>
    <hfp:hasFacet name="whiteSpace"/>
    <hfp:hasProperty name="ordered" value="false"/>
    <hfp:hasProperty name="bounded" value="false"/>
    <hfp:hasProperty name="cardinality" value="countably infinite"/>
    <hfp:hasProperty name="numeric" value="false"/>
  </xs:appinfo>
  <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NOTATION"/>
  <xs:documentation>
    NOTATION cannot be used directly in a schema; rather a type
    must be derived from it by specifying at least one enumeration
    facet whose value is the name of a NOTATION declared in the
    schema.
  </xs:documentation>
</xs:annotation>
<xs:restriction base="xs:anySimpleType">
  <xs:whiteSpace fixed="true" id="NOTATION.whiteSpace" value="collapse"/>
</xs:restriction>
</xs:simpleType>
<xs:annotation>
  <xs:documentation>
    Now the derived primitive types
  </xs:documentation>
</xs:annotation>
<xs:simpleType id="normalizedString" name="normalizedString">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#normalizedString"/>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace id="normalizedString.whiteSpace" value="replace"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="token" name="token">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#token"/>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:whiteSpace id="token.whiteSpace" value="collapse"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="language" name="language">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#language"/>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:pattern id="language.pattern" value="[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*">
      <xs:annotation>
        <xs:documentation source="http://www.ietf.org/rfc/rfc3066.txt">
          pattern specifies the content of section 2.12 of XML 1.0e2
          and RFC 3066 (Revised version of RFC 1766).
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="IDREFS" name="IDREFS">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length"/>
      <hfp:hasFacet name="minLength"/>
      <hfp:hasFacet name="maxLength"/>
      <hfp:hasFacet name="enumeration"/>
      <hfp:hasFacet name="whiteSpace"/>
      <hfp:hasFacet name="pattern"/>
      <hfp:hasProperty name="ordered" value="false"/>
      <hfp:hasProperty name="bounded" value="false"/>
      <hfp:hasProperty name="cardinality" value="countably infinite"/>
      <hfp:hasProperty name="numeric" value="false"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#IDREFS"/>
  </xs:annotation>
  <xs:restriction>
    <xs:simpleType>

```

```

    <xs:list itemType="xs:IDREF" />
  </xs:simpleType>
  <xs:minLength id="IDREFS.minLength" value="1" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType id="ENTITIES" name="ENTITIES">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length" />
      <hfp:hasFacet name="minLength" />
      <hfp:hasFacet name="maxLength" />
      <hfp:hasFacet name="enumeration" />
      <hfp:hasFacet name="whiteSpace" />
      <hfp:hasFacet name="pattern" />
      <hfp:hasProperty name="ordered" value="false" />
      <hfp:hasProperty name="bounded" value="false" />
      <hfp:hasProperty name="cardinality" value="countably infinite" />
      <hfp:hasProperty name="numeric" value="false" />
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#ENTITIES" />
  </xs:annotation>
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:ENTITY" />
    </xs:simpleType>
    <xs:minLength id="ENTITIES.minLength" value="1" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="NMTOKEN" name="NMTOKEN">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NMTOKEN" />
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:pattern id="NMTOKEN.pattern" value="\c+">
      <xs:annotation>
        <xs:documentation source="http://www.w3.org/TR/REC-xml#NT-Nmtoken">
          pattern matches production 7 from the XML spec
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="NMTOKENS" name="NMTOKENS">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length" />
      <hfp:hasFacet name="minLength" />
      <hfp:hasFacet name="maxLength" />
      <hfp:hasFacet name="enumeration" />
      <hfp:hasFacet name="whiteSpace" />
      <hfp:hasFacet name="pattern" />
      <hfp:hasProperty name="ordered" value="false" />
      <hfp:hasProperty name="bounded" value="false" />
      <hfp:hasProperty name="cardinality" value="countably infinite" />
      <hfp:hasProperty name="numeric" value="false" />
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NMTOKENS" />
  </xs:annotation>
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:NMTOKEN" />
    </xs:simpleType>
    <xs:minLength id="NMTOKENS.minLength" value="1" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="Name" name="Name">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#Name" />
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:pattern id="Name.pattern" value="\i\c*">
      <xs:annotation>
        <xs:documentation source="http://www.w3.org/TR/REC-xml#NT-Name">
          pattern matches production 5 from the XML spec
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>

```



```

    </xs:annotation>
  </xs:pattern>
</xs:restriction>
</xs:simpleType>
<xs:simpleType id="NCName" name="NCName">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#NCName"/>
  </xs:annotation>
  <xs:restriction base="xs:Name">
    <xs:pattern id="NCName.pattern" value="[\\i-[:]][\\c-[:]]*">
      <xs:annotation>
        <xs:documentation source="http://www.w3.org/TR/REC-xml-names/#NT-NCName">
          pattern matches production 4 from the Namespaces in XML spec
        </xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="ID" name="ID">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#ID"/>
  </xs:annotation>
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
<xs:simpleType id="IDREF" name="IDREF">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#IDREF"/>
  </xs:annotation>
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
<xs:simpleType id="ENTITY" name="ENTITY">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#ENTITY"/>
  </xs:annotation>
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
<xs:simpleType id="integer" name="integer">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#integer"/>
  </xs:annotation>
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits fixed="true" id="integer.fractionDigits" value="0"/>
    <xs:pattern value="[\\-+]?[0-9]*/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="nonPositiveInteger" name="nonPositiveInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#nonPositiveInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:maxInclusive id="nonPositiveInteger.maxInclusive" value="0"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="negativeInteger" name="negativeInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#negativeInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:nonPositiveInteger">
    <xs:maxInclusive id="negativeInteger.maxInclusive" value="-1"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="long" name="long">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#long"/>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:minInclusive id="long.minInclusive" value="-9223372036854775808"/>
    <xs:maxInclusive id="long.maxInclusive" value="9223372036854775807"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType id="int" name="int">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#int"/>
  </xs:annotation>
  <xs:restriction base="xs:long">
    <xs:minInclusive id="int.minInclusive" value="-2147483648"/>
    <xs:maxInclusive id="int.maxInclusive" value="2147483647"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="short" name="short">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#short"/>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:minInclusive id="short.minInclusive" value="-32768"/>
    <xs:maxInclusive id="short.maxInclusive" value="32767"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="byte" name="byte">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#byte"/>
  </xs:annotation>
  <xs:restriction base="xs:short">
    <xs:minInclusive id="byte.minInclusive" value="-128"/>
    <xs:maxInclusive id="byte.maxInclusive" value="127"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="nonNegativeInteger" name="nonNegativeInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#nonNegativeInteger"/>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:minInclusive id="nonNegativeInteger.minInclusive" value="0"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="unsignedLong" name="unsignedLong">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasProperty name="bounded" value="true"/>
      <hfp:hasProperty name="cardinality" value="finite"/>
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedLong"/>
  </xs:annotation>
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:maxInclusive id="unsignedLong.maxInclusive" value="18446744073709551615"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="unsignedInt" name="unsignedInt">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedInt"/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedLong">
    <xs:maxInclusive id="unsignedInt.maxInclusive" value="4294967295"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="unsignedShort" name="unsignedShort">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedShort"/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedInt">
    <xs:maxInclusive id="unsignedShort.maxInclusive" value="65535"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="unsignedByte" name="unsignedByte">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#unsignedByte"/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedShort">
    <xs:maxInclusive id="unsignedByte.maxInclusive" value="255"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType id="positiveInteger" name="positiveInteger">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#positiveInteger"/>
  </xs:annotation>

```

```

</xs:annotation>
<xs:restriction base="xs:nonNegativeInteger">
  <xs:minInclusive id="positiveInteger.minInclusive" value="1"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="derivationControl">
  <xs:annotation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="substitution"/>
    <xs:enumeration value="extension"/>
    <xs:enumeration value="restriction"/>
    <xs:enumeration value="list"/>
    <xs:enumeration value="union"/>
  </xs:restriction>
</xs:simpleType>
<xs:group name="simpleDerivation">
  <xs:choice>
    <xs:element ref="xs:restriction"/>
    <xs:element ref="xs:list"/>
    <xs:element ref="xs:union"/>
  </xs:choice>
</xs:group>
<xs:simpleType name="simpleDerivationSet">
  <xs:annotation>
    <xs:documentation>
      #all or (possibly empty) subset of {restriction, union, list}
    </xs:documentation>
    <xs:documentation>
      A utility type, not for public use
    </xs:documentation>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="#all"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:list>
        <xs:simpleType>
          <xs:restriction base="xs:derivationControl">
            <xs:enumeration value="list"/>
            <xs:enumeration value="union"/>
            <xs:enumeration value="restriction"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:list>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:complexType abstract="true" name="simpleType">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:group ref="xs:simpleDerivation"/>
      <xs:attribute name="final" type="xs:simpleDerivationSet"/>
      <xs:attribute name="name" type="xs:NCName">
        <xs:annotation>
          <xs:documentation>
            Can be restricted to required or forbidden
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="topLevelSimpleType">
  <xs:complexContent>
    <xs:restriction base="xs:simpleType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:simpleDerivation"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

```

</xs:sequence>
<xs:attribute name="name" type="xs:NCName" use="required">
  <xs:annotation>
    <xs:documentation>
      Required at the top level
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="localSimpleType">
  <xs:complexContent>
    <xs:restriction base="xs:simpleType">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
        <xs:group ref="xs:simpleDerivation"/>
      </xs:sequence>
      <xs:attribute name="name" use="prohibited">
        <xs:annotation>
          <xs:documentation>
            Forbidden when nested
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="final" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:element id="simpleType" name="simpleType" type="xs:topLevelSimpleType">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-simpleType"/>
  </xs:annotation>
</xs:element>
<xs:group name="facets">
  <xs:annotation>
    <xs:documentation>
      We should use a substitution group for facets, but that's ruled out because it would allow users to add their own, which we're not ready for yet.
    </xs:documentation>
  </xs:annotation>
</xs:group>
<xs:choice>
  <xs:element ref="xs:minExclusive"/>
  <xs:element ref="xs:minInclusive"/>
  <xs:element ref="xs:maxExclusive"/>
  <xs:element ref="xs:maxInclusive"/>
  <xs:element ref="xs:totalDigits"/>
  <xs:element ref="xs:fractionDigits"/>
  <xs:element ref="xs:length"/>
  <xs:element ref="xs:minLength"/>
  <xs:element ref="xs:maxLength"/>
  <xs:element ref="xs:enumeration"/>
  <xs:element ref="xs:whiteSpace"/>
  <xs:element ref="xs:pattern"/>
</xs:choice>
</xs:group>
<xs:group name="simpleRestrictionModel">
  <xs:sequence>
    <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
    <xs:group minOccurs="0" maxOccurs="unbounded" minOccurs="0" ref="xs:facets"/>
  </xs:sequence>
</xs:group>
<xs:element id="restriction" name="restriction">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-restriction">
        base attribute and simpleType child are mutually exclusive, but one or other is required
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:annotated">

```

```

    <xs:group ref="xs:simpleRestrictionModel"/>
    <xs:attribute name="base" type="xs:QName" use="optional"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element id="list" name="list">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-list">
itemType attribute and simpleType child are mutually
exclusive, but one or other is required
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:sequence>
          <xs:element minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
        </xs:sequence>
        <xs:attribute name="itemType" type="xs:QName" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="union" name="union">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-union">
memberTypes attribute must be non-empty or there must be
at least one simpleType child
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:annotated">
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="simpleType" type="xs:localSimpleType"/>
        </xs:sequence>
        <xs:attribute name="memberTypes" use="optional">
          <xs:simpleType>
            <xs:list itemType="xs:QName"/>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:complexType name="facet">
  <xs:complexContent>
    <xs:extension base="xs:annotated">
      <xs:attribute name="value" use="required"/>
      <xs:attribute default="false" name="fixed" type="xs:boolean" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="noFixedFacet">
  <xs:complexContent>
    <xs:restriction base="xs:facet">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="fixed" use="prohibited"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:element id="minExclusive" name="minExclusive" type="xs:facet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-minExclusive"/>
  </xs:annotation>
</xs:element>
<xs:element id="minInclusive" name="minInclusive" type="xs:facet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-minInclusive"/>
  </xs:annotation>
</xs:element>

```

```

<xs:element id="maxExclusive" name="maxExclusive" type="xs:facet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-maxExclusive"/>
  </xs:annotation>
</xs:element>
<xs:element id="maxInclusive" name="maxInclusive" type="xs:facet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-maxInclusive"/>
  </xs:annotation>
</xs:element>
<xs:complexType name="numFacet">
  <xs:complexContent>
    <xs:restriction base="xs:facet">
      <xs:sequence>
        <xs:element minOccurs="0" ref="xs:annotation"/>
      </xs:sequence>
      <xs:attribute name="value" type="xs:nonNegativeInteger" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:element id="totalDigits" name="totalDigits">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-totalDigits"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:numFacet">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="value" type="xs:positiveInteger" use="required"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element id="fractionDigits" name="fractionDigits" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-fractionDigits"/>
  </xs:annotation>
</xs:element>
<xs:element id="length" name="length" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-length"/>
  </xs:annotation>
</xs:element>
<xs:element id="minLength" name="minLength" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-minLength"/>
  </xs:annotation>
</xs:element>
<xs:element id="maxLength" name="maxLength" type="xs:numFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-maxLength"/>
  </xs:annotation>
</xs:element>
<xs:element id="enumeration" name="enumeration" type="xs:noFixedFacet">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-enumeration"/>
  </xs:annotation>
</xs:element>
<xs:element id="whiteSpace" name="whiteSpace">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-whiteSpace"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:facet">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="value" use="required">
          <xs:simpleType>

```

```
<xs:restriction base="xs:NMTOKEN">
  <xs:enumeration value="preserve"/>
  <xs:enumeration value="replace"/>
  <xs:enumeration value="collapse"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element id="pattern" name="pattern">
  <xs:annotation>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#element-pattern"/>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:noFixedFacet">
        <xs:sequence>
          <xs:element minOccurs="0" ref="xs:annotation"/>
        </xs:sequence>
        <xs:attribute name="value" type="xs:string" use="required"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Namespace Bindings

Prefix	Namespace URI / Binding Location
-	http://www.w3.org/1999/xhtml File: xml.xsd Element: <code><xs:schema ... ></code> [378]
hfp	http://www.w3.org/2001/XMLSchema-hasFacetAndProperty File: XMLSchema.xsd Element: <code><xs:schema ... ></code> [397]
xs	http://www.w3.org/2001/XMLSchema File: xml.xsd Element: <code><xs:schema ... ></code> [378] File: XMLSchema.xsd Element: <code><xs:schema ... ></code> [397]

WSDL documentation generated with [FlexDoc/XML](#) 1.13 using [FlexDoc/XML WSDLDoc](#) 1.2.5 template set. All XSD diagrams generated by [FlexDoc/XML DiagramKit](#).