Schema documentation for SdumServiceResultSet.xsd

january 24, 2014

Table of Contents

Resource hierarchy:

Legend: *Import*, *Include*, *Redefine*, *Cycle detected*

SdumServiceResultSet.xsd

- protocol-types.xsd
  - result.xsd
    - xml.xsd
  - rdf.xsd

Namespace: "urn:openiot:sdum:serviceresultset:xsd:1"

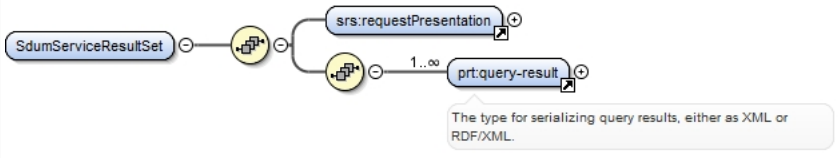
Schema(s)

Main schema `SdumServiceResultSet.xsd`

Namespace	urn:openiot:sdum:serviceresultset:xsd:1
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

Element `srs:SdumServiceResultSet`

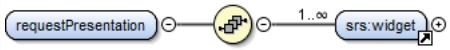
Namespace	urn:openiot:sdum:serviceresultset:xsd:1
Diagram	
Properties	content: `complex`
Model	srs:requestPresentation , st:query-result+
Children	srs:requestPresentation, st:query-result

Instance	`<srs:SdumServiceResultSet xmlns:srs="urn:openiot:sdum:serviceresultset:xsd:1" xmlns:prt="http://www.w3.org/2007/SPARQL/protocol-types#">`
	`  <srs:requestPresentation>{1,1}</srs:requestPresentation>`
	`  <prt:query-result>{1,unbounded}</prt:query-result>`
	`</srs:SdumServiceResultSet>`

Source	`<xs:element name="SdumServiceResultSet">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element ref="srs:requestPresentation" maxOccurs="1"/>`
	`  <xs:sequence>`
	`  <xs:element ref="prt:query-result" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  </xs:sequence>`
	`  </xs:complexType>`

```
</xs:element>
```

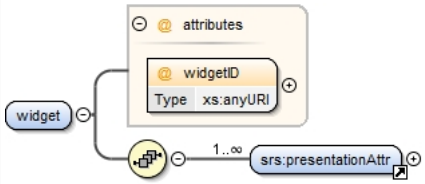
Element `srs:requestPresentation`

Namespace	urn:openiot:sdum:serviceresultset:xsd:1
Diagram	
Properties	content: complex
Used by	Element srs:SdumServiceResultSet
Model	srs:widget+
Children	srs:widget

Instance	`<srs:requestPresentation xmlns:srs="urn:openiot:sdum:serviceresultset:xsd:1">`
	`  <srs:widget widgetID="">{1,unbounded}</srs:widget>`
	`</srs:requestPresentation>`

Source	`<xs:element name="requestPresentation">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element maxOccurs="unbounded" ref="srs:widget"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `srs:widget`

Namespace	urn:openiot:sdum:serviceresultset:xsd:1
Diagram	
Properties	content: complex
Used by	Element srs:requestPresentation
Model	srs:presentationAttr+
Children	srs:presentationAttr

Instance	`<srs:widget widgetID="" xmlns:srs="urn:openiot:sdum:serviceresultset:xsd:1">`
	`  <srs:presentationAttr name="" value="">{1,unbounded}</srs:presentationAttr>`
	`</srs:widget>`

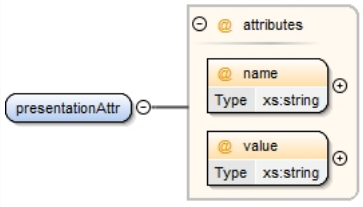
Attributes

QName	Type	Fixed	Default	Use
widgetID	xs:anyURI			required

Source

```
<xs:element name="widget">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="srs:presentationAttr"/>
    </xs:sequence>
    <xs:attribute name="widgetID" use="required" type="xs:anyURI"/>
  </xs:complexType>
</xs:element>
```

Element `srs:presentationAttr`

Namespace	urn:openiot:sdum:serviceresultset:xsd:1
Diagram	

Properties	content:	complex
Used by	Element	srs:widget

Attributes

QName	Type	Fixed	Default	Use
name	xs:string			required
value	xs:string			required

Source

```
<xs:element name="presentationAttr">
  <xs:complexType>
    <xs:attribute name="name" use="required" type="xs:string"/>
    <xs:attribute name="value" use="required" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

Namespace: "http://www.w3.org/2007/SPARQL/protocol-types#"

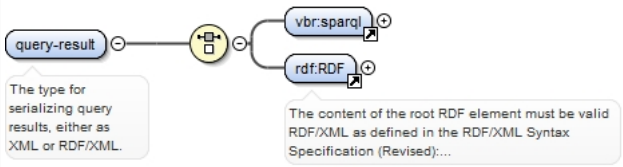
Schema(s)

Imported schema `protocol-types.xsd`

Namespace	http://www.w3.org/2007/SPARQL/protocol-types#	
Properties	attribute form default:	unqualified
	element form default:	unqualified
	version:	$Id: protocol-types.xsd,v 1.2 2007/10/17 17:17:35 eric Exp $

Element(s)

Element `st:query-result`

Namespace	http://www.w3.org/2007/SPARQL/protocol-types#	
Annotations	The type for serializing query results, either as XML or RDF/XML.	
Diagram		
Properties	content:	complex
Used by	Element	srs:SdumServiceResultSet
Model	res:sparql | rdf:RDF	
Children	rdf:RDF, res:sparql	

Instance

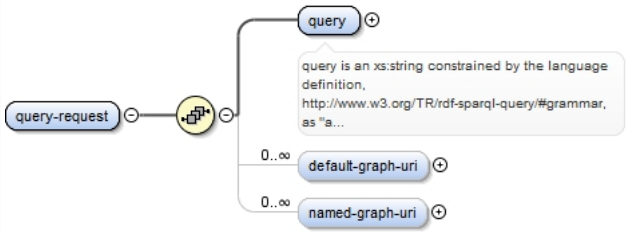
```
<st:query-result xmlns:st="http://www.w3.org/2007/SPARQL/protocol-types#" xmlns:vbr="http://
www.w3.org/2007/SPARQL/results#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <vbr:sparql>{1,1}</vbr:sparql>
  <rdf:RDF>{1,1}</rdf:RDF>
</st:query-result>
```

Source

```
<xs:element name="query-result">
  <xs:annotation>
    <xs:documentation>The type for serializing query results, either as XML or RDF/XML.</
xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice>
      <xs:element maxOccurs="1" ref="vbr:sparql"/>
      <xs:element maxOccurs="1" ref="rdf:RDF"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Element `st:query-request`

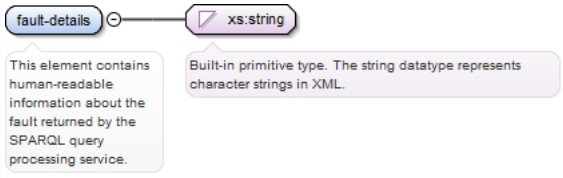
Namespace	http://www.w3.org/2007/SPARQL/protocol-types#

Diagram	
Properties	content: complex
Model	query , default-graph-uri* , named-graph-uri*
Children	default-graph-uri, named-graph-uri, query

Instance	`<st:query-request xmlns:st="http://www.w3.org/2007/SPARQL/protocol-types#">`
	`  <query>{1,1}</query>`
	`  <default-graph-uri>{0,unbounded}</default-graph-uri>`
	`  <named-graph-uri>{0,unbounded}</named-graph-uri>`
	`</st:query-request>`

Source	`<xs:element name="query-request">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element minOccurs="1" maxOccurs="1" name="query" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>query is an xs:string constrained by the language definition, http://www.w3.org/TR/rdf-sparql-query/#grammar, as "a sequence of characters in the language defined by the [SPARQL] grammar, starting with the Query production"</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element minOccurs="0" maxOccurs="unbounded" name="default-graph-uri" type="xs:anyURI"/>`
	`  <xs:element minOccurs="0" maxOccurs="unbounded" name="named-graph-uri" type="xs:anyURI"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element st:fault-details

Namespace	http://www.w3.org/2007/SPARQL/protocol-types#
Annotations	This element contains human-readable information about the fault returned by the SPARQL query processing service.
Diagram	
Type	xs:string
Properties	content: simple
Used by	Elements st:malformed-query, st:query-request-refused

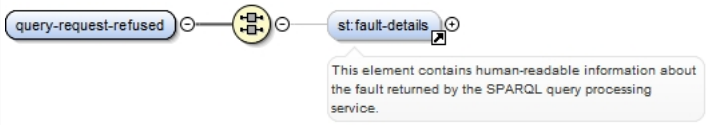
Source	`<xs:element type="xs:string" name="fault-details">`
	`  <xs:annotation>`
	`  <xs:documentation>This element contains human-readable information about the fault returned by the SPARQL query processing service.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element st:malformed-query

Namespace	http://www.w3.org/2007/SPARQL/protocol-types#
Diagram	
Properties	content: complex
Model	ALL(st:fault-details{0,1})

Children	st:fault-details
Instance	<st:malformed-query xmlns:st="http://www.w3.org/2007/SPARQL/protocol-types#"> <st:fault-details>{0,1}</st:fault-details> </st:malformed-query>
Source	<xs:element name="malformed-query"> <xs:complexType> <xs:all> <xs:element minOccurs="0" maxOccurs="1" ref="st:fault-details"/> </xs:all> </xs:complexType> </xs:element>

Element `st:query-request-refused`

Namespace	http://www.w3.org/2007/SPARQL/protocol-types#
Diagram	
Properties	content: complex
Model	ALL(st:fault-details{0,1})
Children	st:fault-details
Instance	<st:query-request-refused xmlns:st="http://www.w3.org/2007/SPARQL/protocol-types#"> <st:fault-details>{0,1}</st:fault-details> </st:query-request-refused>
Source	<xs:element name="query-request-refused"> <xs:complexType> <xs:all> <xs:element minOccurs="0" maxOccurs="1" ref="st:fault-details"/> </xs:all> </xs:complexType> </xs:element>

Namespace: "http://www.w3.org/2007/SPARQL/results#"

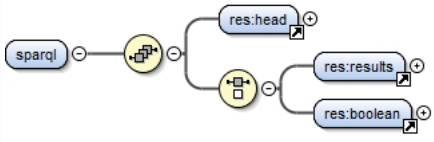
Schema(s)

Imported schema `result.xsd`

Namespace	http://www.w3.org/2007/SPARQL/results#	
Properties	attribute form default:	unqualified
	element form default:	qualified

Element(s)

Element `res:sparql`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Properties	content: complex
Used by	Element st:query-result
Model	res:head , (res:results | res:boolean)
Children	res:boolean, res:head, res:results
Instance	<res:sparql xmlns:res="http://www.w3.org/2007/SPARQL/results#"> <res:head>{1,1}</res:head>

	<pre> <res:results>{1,1}</res:results>\n <res:boolean>{1,1}</res:boolean>\n</res:sparql></pre>
Source	<pre><xs:element name="sparql">\n <xs:complexType>\n <xs:sequence>\n <xs:element ref="res:head"/>\n <xs:choice>\n <xs:element ref="res:results"/>\n <xs:element ref="res:boolean"/>\n </xs:choice>\n </xs:sequence>\n </xs:complexType>\n</xs:element></pre>

Element `res:head`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Properties	content: complex
Used by	Element res:sparql
Model	res:variable* , res:link*
Children	res:link, res:variable
Instance	<pre><res:head xmlns:res="http://www.w3.org/2007/SPARQL/results#">\n <res:variable name="">{0,unbounded}</res:variable>\n <res:link href="">{0,unbounded}</res:link>\n</res:head></pre>
Source	<pre><xs:element name="head">\n <xs:complexType>\n <xs:sequence>\n <xs:element minOccurs="0" maxOccurs="unbounded" ref="res:variable"/>\n <xs:element minOccurs="0" maxOccurs="unbounded" ref="res:link"/>\n </xs:sequence>\n </xs:complexType>\n</xs:element></pre>

Element `res:variable`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Properties	content: complex
Used by	Element res:head

Attributes

QName	Type	Fixed	Default	Use
name	xs:NMTOKEN			required

Source

<pre><xs:element name="variable">\n <xs:complexType>\n   <xs:attributeGroup ref="res:nameAttr"/>\n </xs:complexType>\n</xs:element></pre>

Element `res:link`

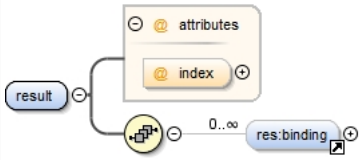
Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	

Properties	content:	complex			
Used by	Element	res:head			
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	href	res:URI-reference			required
Source	<xs:element name="link"> <xs:complexType> <xs:attributeGroup ref="res:hrefAttr"/> </xs:complexType> </xs:element>				

Element res:results

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Properties	content: complex
Used by	Element res:sparql
Model	res:result*
Children	res:result
Instance	<res:results xmlns:res="http://www.w3.org/2007/SPARQL/results#"> <res:result index="">{0,unbounded}</res:result> </res:results>
Source	<xs:element name="results"> <xs:complexType> <xs:sequence> <xs:element minOccurs="0" maxOccurs="unbounded" ref="res:result"/> </xs:sequence> </xs:complexType> </xs:element>

Element res:result

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Properties	content: complex
Used by	Element res:results
Model	res:binding*
Children	res:binding
Instance	<res:result index="" xmlns:res="http://www.w3.org/2007/SPARQL/results#"> <res:binding name="">{0,unbounded}</res:binding> </res:result>

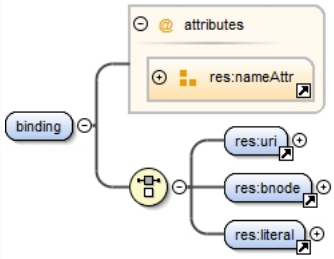
Attributes	QName	Type	Fixed	Default	Use
	index	xs:positiveInteger			optional

Source

```
<xs:element name="result">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="res:binding"/>
    </xs:sequence>
    <xs:attribute name="index" type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
```

Element res:binding

Namespace	http://www.w3.org/2007/SPARQL/results#

Diagram	
Properties	content: complex
Used by	Element res:result
Model	res:uri | res:bnode | res:literal
Children	res:bnode, res:literal, res:uri

Instance	`<res:binding name="" xmlns:res="http://www.w3.org/2007/SPARQL/results#">`
	`  <res:uri>{1,1}</res:uri>`
	`  <res:bnode>{1,1}</res:bnode>`
	`  <res:literal datatype="" xml:lang="">{1,1}</res:literal>`
	`</res:binding>`

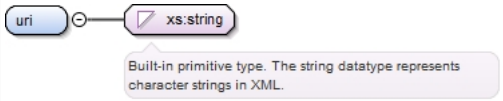
Attributes

QName	Type	Fixed	Default	Use
name	xs:NMTOKEN			required

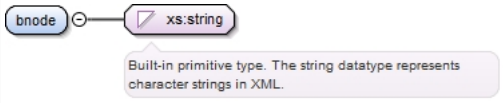
Source

```
<xs:element name="binding">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="res:uri"/>
      <xs:element ref="res:bnode"/>
      <xs:element ref="res:literal"/>
    </xs:choice>
    <xs:attributeGroup ref="res:nameAttr"/>
  </xs:complexType>
</xs:element>
```

Element res:uri

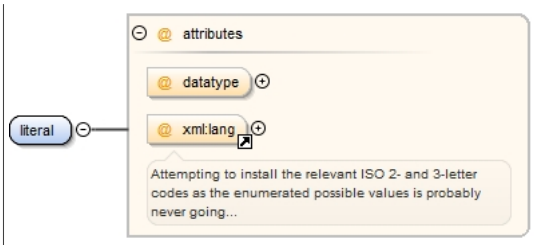
Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Type	xs:string
Properties	content: simple
Used by	Element res:binding
Source	`<xs:element name="uri" type="xs:string"/>`

Element res:bnode

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Type	xs:string
Properties	content: simple
Used by	Element res:binding
Source	`<xs:element name="bnode" type="xs:string"/>`

Element res:literal

Namespace	http://www.w3.org/2007/SPARQL/results#

Diagram	
Properties	content: `complex` mixed: `true`
Used by	Element res:binding
Model	

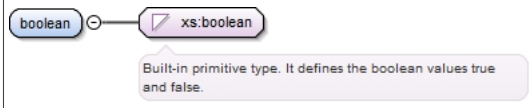
Attributes

QName	Type	Fixed	Default	Use
datatype	res:URI-reference			optional
xml:lang	union of(xs:language, restriction of xs:string)			optional
	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 RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			

Source

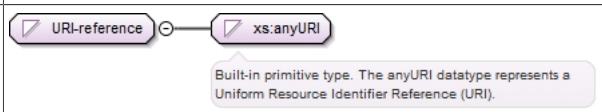
```
<xs:element name="literal">
  <xs:complexType mixed="true">
    <xs:attribute name="datatype" type="res:URI-reference"/>
    <xs:attribute ref="xml:lang"/>
  </xs:complexType>
</xs:element>
```

Element `res:boolean`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Type	xs:boolean
Properties	content: `simple`
Used by	Element res:sparql
Source	`<xs:element name="boolean" type="xs:boolean"/>`

Simple Type(s)

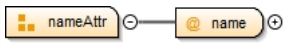
Simple Type `res:URI-reference`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Type	xs:anyURI
Used by	Attributes res:datatypeAttr/@datatype, res:hrefAttr/@href, res:literal/@datatype

Source	`<xs:simpleType name="URI-reference">`
	`  <xs:restriction base="xs:anyURI"/>`
	`</xs:simpleType>`

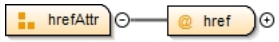
Attribute Group(s)

Attribute Group `res:nameAttr`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Used by	Elements res:binding, res:variable
Attributes	**QName** **Type** **Fixed** **Default** **Use** **name** xs:NMTOKEN required

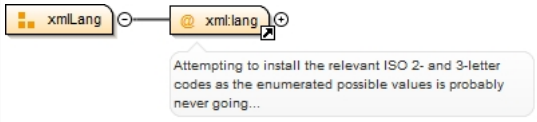
Source	`<xs:attributeGroup name="nameAttr">`
	`  <xs:attribute name="name" use="required" type="xs:NMTOKEN"/>`
	`</xs:attributeGroup>`

Attribute Group `res:hrefAttr`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Used by	Element res:link
Attributes	**QName** **Type** **Fixed** **Default** **Use** **href** res:URI-reference required

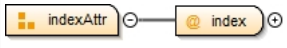
Source	`<xs:attributeGroup name="hrefAttr">`
	`  <xs:attribute name="href" use="required" type="res:URI-reference"/>`
	`</xs:attributeGroup>`

Attribute Group `res:xmlLang`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Attributes	**QName** **Type** **Fixed** **Default** **Use** **xml:lang** union of(xs:language, restriction of xs:string) required `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 RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information.` `The union allows for the 'un-declaration' of xml:lang with the empty string.`

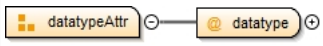
Source	`<xs:attributeGroup name="xmlLang">`
	`  <xs:attribute ref="xml:lang" use="required"/>`
	`</xs:attributeGroup>`

Attribute Group `res:indexAttr`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	
Attributes	**QName** **Type** **Fixed** **Default** **Use** **index** xs:positiveInteger required
Source	`<xs:attributeGroup name="indexAttr">` `  <xs:attribute name="index" use="required" type="xs:positiveInteger"/>`

```
</xs:attributeGroup>
```

Attribute Group `res:datatypeAttr`

Namespace	http://www.w3.org/2007/SPARQL/results#
Diagram	

Attributes

QName	Type	Fixed	Default	Use
datatype	res:URI-reference			required

Source

```
<xs:attributeGroup name="datatypeAttr">
  <xs:attribute name="datatype" use="required" type="res:URI-reference"/>
</xs:attributeGroup>
```

Namespace: "http://www.w3.org/XML/1998/namespace"

Schema(s)

Imported schema `xml.xsd`

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	See http://www.w3.org/XML/1998/namespace.html and http://www.w3.org/TR/REC-xml for information about this namespace. This schema document describes the XML namespace, in a form suitable for import by other schema documents. Note that local names in this namespace are intended to be defined only by the World Wide Web Consortium or its subgroups. The following names are currently defined in this namespace and should not be used with conflicting semantics by any Working Group, specification, or document instance: 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. 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. 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. 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. 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 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"/>

	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 In keeping with the XML Schema WG's standard versioning policy, this schema document will persist at http://www.w3.org/2007/08/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/2007/08/xml.xsd will not change.
Properties	attribute form default: unqualified element form default: unqualified

Attribute(s)

Attribute @xml:lang

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	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 RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.
Type	union of(xs:language, restriction of xs:string)
Properties	content: simple
Used by	Element: res:literal Attribute Groups: res:xmlLang, xml:specialAttrs
Source	<xs:attribute name="lang"> <xs:annotation> <xs:documentation>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 RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.</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	restriction of xs:NCName
Properties	content: simple
Facets	enumeration: default enumeration: preserve

Used by	Attribute Group xml:specialAttrs
Source	<xs:attribute name="space"> <xs:simpleType> <xs:restriction base="xs:NCName"> <xs:enumeration value="default"/> <xs:enumeration value="preserve"/> </xs:restriction> </xs:simpleType> </xs:attribute>

Attribute `@xml:base`

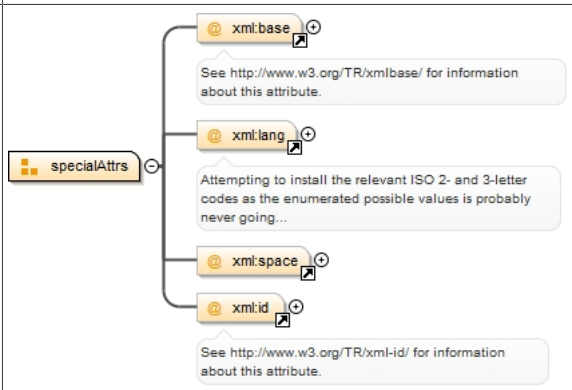
Namespace	http://www.w3.org/XML/1998/namespace
Annotations	See http://www.w3.org/TR/xmlbase/ for information about this attribute.
Type	xs:anyURI
Properties	content: simple
Used by	Attribute Group xml:specialAttrs
Source	<xs:attribute name="base" type="xs:anyURI"> <xs:annotation> <xs:documentation>See http://www.w3.org/TR/xmlbase/ for information about this attribute.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `@xml:id`

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	See http://www.w3.org/TR/xml-id/ for information about this attribute.
Type	xs:ID
Properties	content: simple
Used by	Attribute Group xml:specialAttrs
Source	<xs:attribute name="id" type="xs:ID"> <xs:annotation> <xs:documentation>See http://www.w3.org/TR/xml-id/ for information about this attribute.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute Group(s)

Attribute Group `xml:specialAttrs`

Namespace	http://www.w3.org/XML/1998/namespace
Diagram	

Attributes

QName	Type	Fixed	Default	Use
xml:base	xs:anyURI			optional
	See http://www.w3.org/TR/xmlbase/ for			

	QName	Type	Fixed	Default	Use
		information about this attribute.			
	xml:id	xs:ID			optional
		See http://www.w3.org/TR/xml-id/ for information about this attribute.			
	xml:lang	union of(xs:language, restriction of xs:string)			optional
		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 RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			
	xml:space	restriction of xs:NCName			optional

Source

```
<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>
```

Namespace: "http://www.w3.org/1999/02/22-rdf-syntax-ns#"

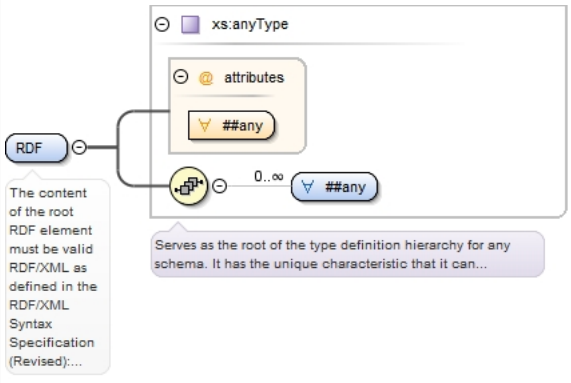
Schema(s)

Imported schema `rdf.xsd`

Namespace	http://www.w3.org/1999/02/22-rdf-syntax-ns#	
Properties	attribute form default:	unqualified
	element form default:	unqualified
	version:	$Id: rdf.xsd,v 1.1 2007/10/17 14:48:47 eric Exp $

Element(s)

Element `rdf:RDF`

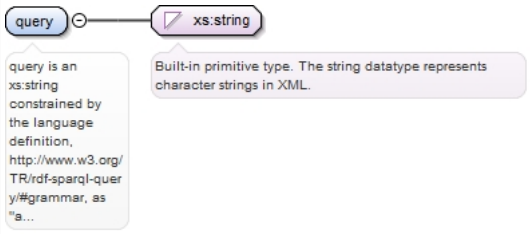
Namespace	http://www.w3.org/1999/02/22-rdf-syntax-ns#
Annotations	The content of the root RDF element must be valid RDF/XML as defined in the RDF/XML Syntax Specification (Revised): http://www.w3.org/TR/rdf-syntax-grammar/
Diagram	
Used by	Element st:query-result	
Source	<xs:element name="RDF" type="xs:anyType"> <xs:annotation> <xs:documentation xml:lang="en">The content of the root RDF element must be valid RDF/XML as defined in the RDF/XML Syntax Specification (Revised): http://www.w3.org/TR/rdf-syntax-grammar/</xs:documentation>	

```
  </xs:annotation>
</xs:element>
```

Namespace: ""

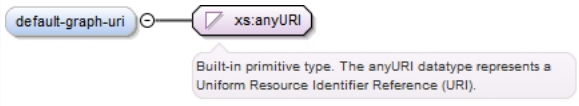
Element(s)

Element st:query-request / query

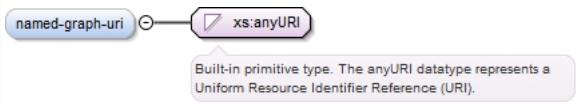
Namespace	No namespace
Annotations	query is an xs:string constrained by the language definition, http://www.w3.org/TR/rdf-sparql-query/#grammar, as "a sequence of characters in the language defined by the [SPARQL] grammar, starting with the Query production"
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 1 maxOccurs: 1

Source	`<xs:element minOccurs="1" maxOccurs="1" name="query" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>query is an xs:string constrained by the language definition, http://www.w3.org/TR/rdf-sparql-query/#grammar, as "a sequence of characters in the language defined by the [SPARQL] grammar, starting with the Query production"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element st:query-request / default-graph-uri

Namespace	No namespace
Diagram	
Type	xs:anyURI
Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Source	`<xs:element minOccurs="0" maxOccurs="unbounded" name="default-graph-uri" type="xs:anyURI"/>`

Element st:query-request / named-graph-uri

Namespace	No namespace
Diagram	
Type	xs:anyURI
Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Source	`<xs:element minOccurs="0" maxOccurs="unbounded" name="named-graph-uri" type="xs:anyURI"/>`

Attribute(s)

Attribute `srs:presentationAttr / @name`

Namespace	No namespace
Type	xs:string
Properties	use: `required`
Used by	Element srs:presentationAttr
Source	`<xs:attribute name="name" use="required" type="xs:string"/>`

Attribute `srs:presentationAttr / @value`

Namespace	No namespace
Type	xs:string
Properties	use: `required`
Used by	Element srs:presentationAttr
Source	`<xs:attribute name="value" use="required" type="xs:string"/>`

Attribute `srs:widget / @widgetID`

Namespace	No namespace
Type	xs:anyURI
Properties	use: `required`
Used by	Element srs:widget
Source	`<xs:attribute name="widgetID" use="required" type="xs:anyURI"/>`

Attribute `res:nameAttr / @name`

Namespace	No namespace
Type	xs:NMTOKEN
Properties	use: `required`
Used by	Attribute Group res:nameAttr
Source	`<xs:attribute name="name" use="required" type="xs:NMTOKEN"/>`

Attribute `res:hrefAttr / @href`

Namespace	No namespace
Type	res:URI-reference
Properties	use: `required`
Used by	Attribute Group res:hrefAttr
Source	`<xs:attribute name="href" use="required" type="res:URI-reference"/>`

Attribute `res:literal / @datatype`

Namespace	No namespace
Type	res:URI-reference
Properties	content: `simple`
Used by	Element res:literal
Source	`<xs:attribute name="datatype" type="res:URI-reference"/>`

Attribute `res:result / @index`

Namespace	No namespace

Type	xs:positiveInteger
Properties	content: simple
Used by	Element res:result
Source	`<xs:attribute name="index" type="xs:positiveInteger"/>`

Attribute `res:indexAttr / @index`

Namespace	No namespace
Type	xs:positiveInteger
Properties	use: required
Used by	Attribute Group res:indexAttr
Source	`<xs:attribute name="index" use="required" type="xs:positiveInteger"/>`

Attribute `res:datatypeAttr / @datatype`

Namespace	No namespace
Type	res:URI-reference
Properties	use: required
Used by	Attribute Group res:datatypeAttr
Source	`<xs:attribute name="datatype" use="required" type="res:URI-reference"/>`