

# XML Schema for WSMML identifiers

---

## Table of Contents

- [Schema Document Properties](#)
- [Global Schema Components](#)
  - [Simple Type: \*\*wsmIIIRI\*\*](#)
  - [Complex Type: \*\*wsmIAnyValue\*\*](#)
  - [Complex Type: \*\*wsmIID\*\*](#)
  - [Simple Type: \*\*wsmIVariable\*\*](#)

[top](#)

---

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.wsmo.org/wsmml/wsmml-syntax#">http://www.wsmo.org/wsmml/wsmml-syntax#</a>
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"><li>• Global element and attribute declarations belong to this schema's target namespace.</li><li>• By default, local element declarations belong to this schema's target namespace.</li><li>• By default, local attribute declarations have no namespace.</li></ul>
<b>Documentation</b>	version: \$Revision: 1.13 \$ date: \$Date: 2005/03/04 11:50:15 \$ author: Jos de Bruijn this schema is a module, which belongs to the WSMML/XML schema specification. This schema provides the necessary definitions for the identifiers in WSMML/XML.

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.wsmo.org/wsmml/wsmml-syntax#">http://www.wsmo.org/wsmml/wsmml-syntax#</a>
xml	http://www.w3.org/XML/1998/namespace
xs	http://www.w3.org/2001/XMLSchema

### Schema Component Representation

```
<xs:schema targetNamespace="http://www.wsmo.org/wsmml/wsmml-syntax#"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  ...
</xs:schema>
```

[top](#)

---

## Global Schema Components

Simple Type: **wsmIIIRI**

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

**Name** wsmlIRI

**Content**

- Union of following types:
  - [xs:string](#)

**Documentation**

The basic kind of identifier in WSML: an IRI. As an URI is more restrictive than an IRI, we are forced to use `xs:string` instead of `xs:anyURI` as type! Note that when translating the standard WSML syntax to XML, Qualified names are resolved to full IRIs. When using the reserved IRI `http://www.wsmo.org/2004/wsml#anonymousID`, it is interpreted as an anonymous ID.

**Schema Component Representation**

```
<xs:simpleType name="wsmlIRI">
  <xs:union memberTypes=" xs:string"/>
</xs:simpleType>
```

[top](#)

---

## Complex Type: **wsmlAnyValue**

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

**Name** wsmlAnyValue

**Abstract**

no

**Documentation**

A data value in WSML. The type is required. Because we allow complex constructed data values, a data value can have multiple arguments. The arguments are strictly ordered. The content is mixed; it is thus possible to use both regular text and tags which represent the arguments. Regular text is interpreted as the first argument.

**XML Instance Representation**

```
<...
  type=" wsmlIRI [1]">
  <!-- Mixed content -->
  <argument> xs:string </argument> [0..*]
</...>
```

**Schema Component Representation**

```
<xs:complexType name="wsmlAnyValue" mixed="true">
  <xs:sequence>
    <xs:element name="argument" type=" xs:string " minOccurs="0"
```

```

        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type=" wsm1IRI " use="required"/>
</xs:complexType>

```

[top](#)

## Complex Type: **wsm1ID**

Super-types: [xs:string](#) < **wsm1ID** (by extension)  
Sub-types: None

<b>Name</b>	wsm1ID
<b>Abstract</b>	no
<b>Documentation</b>	wsm1ID corresponds with either an IRI, a string, an integer, a decimal, a variable or an anonymous identifier. The type of the identifier is indicated in the attribute 'type'. The IRIs identifying the types IRI, string, integer and decimal correspond with the IRIs of the datatypes as defined in Appendix C of the WSM1 specification. The type variable is identified with the IRI <a href="http://www.wsmo.org/2004/wsm1#variable">http://www.wsmo.org/2004/wsm1#variable</a> , the type anonymousID is identified with <a href="http://www.wsmo.org/2004/wsm1#anonymousID">http://www.wsmo.org/2004/wsm1#anonymousID</a> .

### XML Instance Representation

```

<...
  type=" wsm1IRI [1]">
    xs:string
</...>

```

### Schema Component Representation

```

<xs:complexType name="wsm1ID">
  <xs:simpleContent>
    <xs:extension base=" xs:string ">
      <xs:attribute name="type" type=" wsm1IRI " use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

[top](#)

## Simple Type: **wsm1Variable**

*Super-types:*        [xs:string](#) < **wsmIVariable** (by restriction)

*Sub-types:*         None

**Name**                                wsmIVariable

**Content**

- Base XSD Type: string
- *pattern* = `'?([a-z][A-Z][0-9][0x4E00 - 0x9FA5][0x3007][0x3021 - 0x3029])+'`

**Documentation**                      A variables in a WSM logical expression.

#### Schema Component Representation

```
<xs:simpleType name="wsmIVariable">
  <xs:restriction base=" xs:string ">
    <xs:pattern value="'?' ([a-z] [A-Z] [0-9] [0x4E00 -
      0x9FA5] [0x3007] [0x3021 - 0x3029])+'"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

---

Generated by [xs3p](#).