Schema documentation for serializable_schema.xsd

august 31, 2016

Table of Contents

Namespace: ""

Schema(s)

Main schema `serializable_schema.xsd`

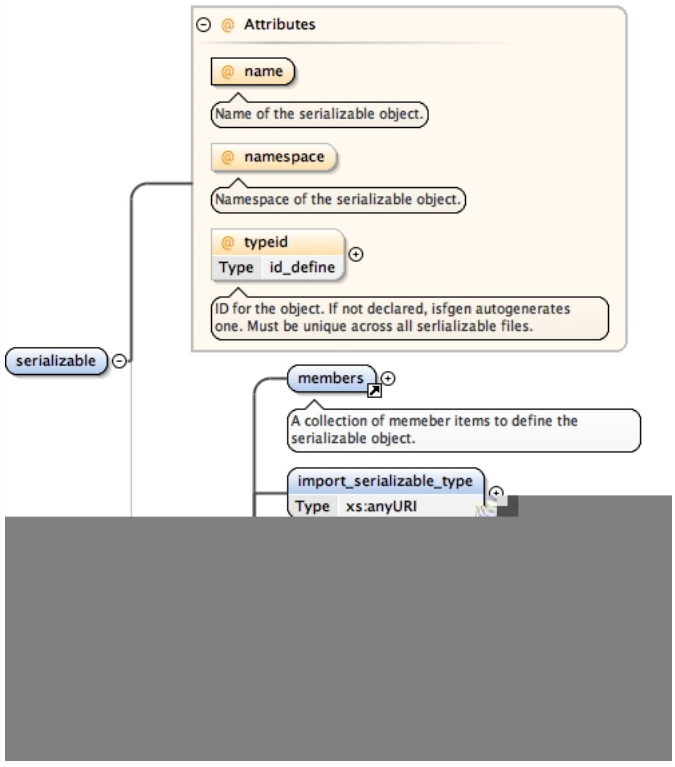
Namespace	No namespace
Properties	attribute form default: `unqualified` element form default: `qualified`

Included schema `common_elements.xsd`

Namespace	No namespace
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

Element `serializable`

Namespace	No namespace
Diagram	
Properties	content: `complex`
Model	members | import_serializable_type | include_header | comment
Children	comment, import_serializable_type, include_header, members

Instance	`<serializable name="" namespace="" typeid="">`
	`  <members>{1,1}</members>`
	`  <import_serializable_type>{1,1}</import_serializable_type>`
	`  <include_header>{1,1}</include_header>`
	`  <comment>{1,1}</comment>`
	`</serializable>`

Attributes

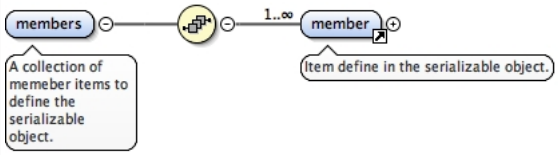
QName	Type	Use	
name		required	

	QName	Type	Use	
		Name of the serializable object.		
	namespace		optional	
		Namespace of the serializable object.		
	typeid	id_define	optional	
		ID for the object. If not declared, isfgen autogenerates one. Must be unique across all serlializable files.		

Source

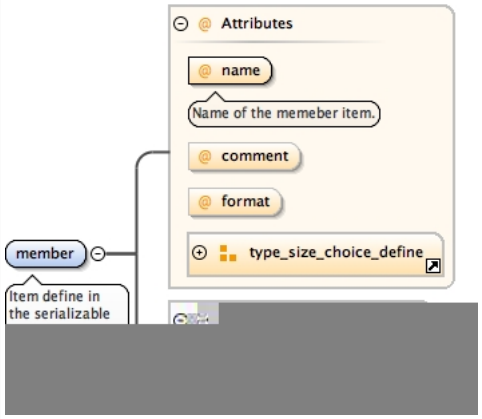
```
<xs:element name="serializable">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="members"/>
      <xs:element ref="import_serializable_type"/>
      <xs:element ref="include_header"/>
      <xs:element ref="comment"/>
    </xs:choice>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the serializable object.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="namespace">
      <xs:annotation>
        <xs:documentation>Namespace of the serializable object.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="typeid" type="id_define">
      <xs:annotation>
        <xs:documentation>ID for the object. If not declared, isfgen autogenerates one. Must be
 unique across all serlializable files.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element `members`

Namespace	No namespace
Annotations	A collection of memeber items to define the serializable object.
Diagram	
Properties	content: complex
Used by	Element serializable
Model	member+
Children	member
Instance	`<members> <member comment="" data_type="" format="" name="" size="" type="">{1,unbounded}</member> </members>`
Source	`<xs:element name="members"> <xs:annotation> <xs:documentation>A collection of memeber items to define the serializable object.</ xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element maxOccurs="unbounded" ref="member"/> </xs:sequence> </xs:complexType> </xs:element>`

Element `member`

Namespace	No namespace
Annotations	Item define in the serializable object.

Diagram	
Properties	content: complex
Used by	Element members
Model	enum{0,1}
Children	enum

Instance	`<member comment="" data_type="" format="" name="" size="" type="">`
	`  <enum name="">{0,1}</enum>`
	`</member>`

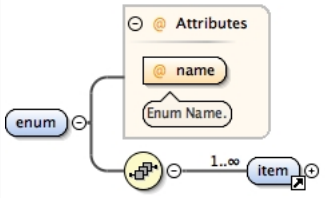
Attributes

QName	Type	Use	
comment		optional	
data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
format		optional	
name		required	
	Name of the memeber item.		
size	xs:nonNegativeInteger	optional	
	The size of the argument.		
type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	

Source

```
<xs:element name="member">
  <xs:annotation>
    <xs:documentation>Item define in the serializable object.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:group ref="type_size_choice_define"/>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the memeber item.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="comment"/>
    <xs:attribute name="format"/>
    <xs:attributeGroup ref="type_size_choice_define"/>
  </xs:complexType>
</xs:element>
```

Element enum

Namespace	No namespace
Diagram	

Properties	content:	complex
Used by	Element Group	type_size_choice_define
Model	item+	
Children	item	
Instance	<enum name="">
<item comment="" name="" value="">{1,unbounded}</item>
</enum>	

Attributes	QName	Type	Use	
	name		required	
		Enum Name.		

Source

```
<xs:element name="enum">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="item"/>
    </xs:sequence>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Enum Name.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element item

Namespace	No namespace
Diagram	
Properties	content: complex
Used by	Element enum

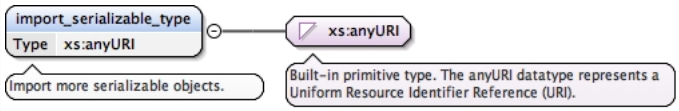
Attributes	QName	Type	Use	
	comment		optional	
		Comment about the enum item.		
	name		required	
		Name of the enum item.		
	value		optional	
		The value being sent through the enum item.		

Source

```
<xs:element name="item">
  <xs:complexType>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the enum item.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="value">
      <xs:annotation>
        <xs:documentation>The value being sent through the enum item.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="comment">
      <xs:annotation>
        <xs:documentation>Comment about the enum item.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
```

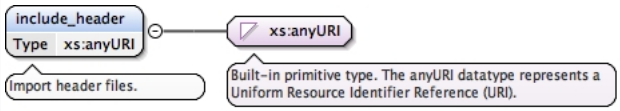
```
  </xs:complexType>
</xs:element>
```

Element `import_serializable_type`

Namespace	No namespace
Annotations	`Import more serializable objects.`
Diagram	
Type	xs:anyURI
Properties	content: `simple`
Used by	Element serializable

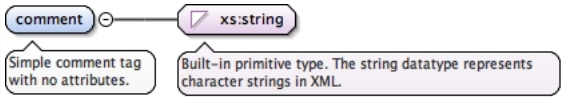
Source	`<xs:element name="import_serializable_type" type="xs:anyURI">`
	`  <xs:annotation>`
	`  <xs:documentation>Import more serializable objects.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `include_header`

Namespace	No namespace
Annotations	`Import header files.`
Diagram	
Type	xs:anyURI
Properties	content: `simple`
Used by	Element serializable

Source	`<xs:element name="include_header" type="xs:anyURI">`
	`  <xs:annotation>`
	`  <xs:documentation>Import header files.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `comment`

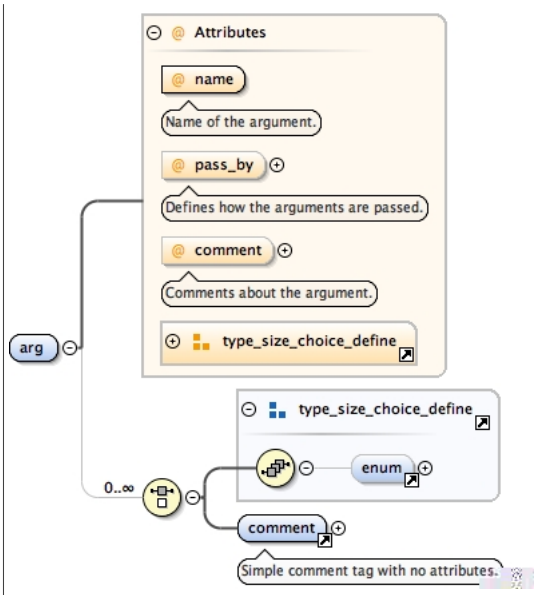
Namespace	No namespace
Annotations	`Simple comment tag with no attributes.`
Diagram	
Type	xs:string
Properties	content: `simple`
Used by	Elements arg_define/arg, external_arg_define/arg, return, serializable

Source	`<xs:element name="comment" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>Simple comment tag with no attributes.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `arg_define / arg`

Namespace	No namespace

Diagram



Properties

content: complex

Model

(enum{0,1}) | comment

Children

comment, enum

Instance

```
<arg comment="" data_type="" name="" pass_by="" size="" type="">
  <enum name="">{0,1}</enum>
  <comment>{1,1}</comment>
</arg>
```

Attributes

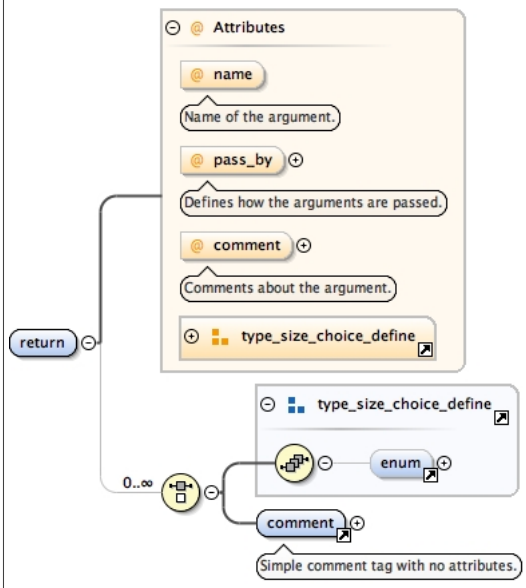
QName	Type	Use	
comment	xs:string	optional	
	Comments about the argument.		
data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
name		required	
	Name of the argument.		
pass_by	pass_by_define	optional	
	Defines how the arguments are passed.		
size	xs:nonNegativeInteger	optional	
	The size of the argument.		
type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	

Source

```
<xs:element name="arg">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="type_size_choice_define"/>
      <xs:element ref="comment"/>
    </xs:choice>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the argument.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pass_by" type="pass_by_define">
      <xs:annotation>
        <xs:documentation>Defines how the arguments are passed.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="comment" type="xs:string">
      <xs:annotation>
        <xs:documentation>Comments about the argument.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="type_size_choice_define"/>
  </xs:complexType>
```

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

Element `return`

Namespace	No namespace
Diagram	
Properties	content: complex
Model	(enum{0,1}) | comment
Children	comment, enum

Instance	`<return comment="" data_type="" name="" pass_by="" size="" type="">`
	`  <enum name="">{0,1}</enum>`
	`  <comment>{1,1}</comment>`
	`</return>`

Attributes

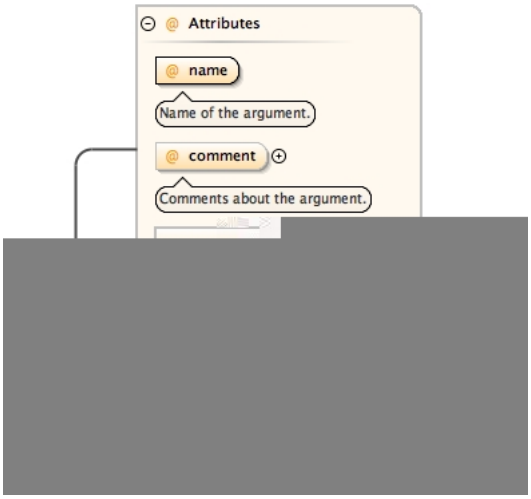
QName	Type	Use	
comment	xs:string	optional	
	Comments about the argument.		
data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
name		optional	
	Name of the argument.		
pass_by	pass_by_define	optional	
	Defines how the arguments are passed.		
size	xs:nonNegativeInteger	optional	
	The size of the argument.		
type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	

Source

```
<xs:element name="return">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="type_size_choice_define"/>
      <xs:element ref="comment"/>
    </xs:choice>
    <xs:attribute name="name">
      <xs:annotation>
        <xs:documentation>Name of the argument.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pass_by" type="pass_by_define">
      <xs:annotation>
        <xs:documentation>Defines how the arguments are passed.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="comment" type="xs:string">
```

```
      <xs:annotation>
        <xs:documentation>Comments about the argument.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="type_size_choice_define"/>
  </xs:complexType>
</xs:element>
```

Element `external_arg_define / arg`

Namespace	No namespace
Diagram	
Properties	content: complex
Model	(enum{0,1}) | comment
Children	comment, enum

Instance	`<arg comment="" data_type="" name="" size="" type="">`
	`  <enum name="">{0,1}</enum>`
	`  <comment>{1,1}</comment>`
	`</arg>`

Attributes

QName	Type	Use	
comment	xs:string	optional	
	Comments about the argument.		
data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
name		required	
	Name of the argument.		
size	xs:nonNegativeInteger	optional	
	The size of the argument.		
type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	

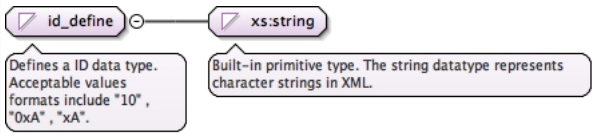
Source

```
<xs:element name="arg">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="type_size_choice_define"/>
      <xs:element ref="comment"/>
    </xs:choice>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the argument.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="comment" type="xs:string">
      <xs:annotation>
        <xs:documentation>Comments about the argument.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="type_size_choice_define"/>
  </xs:complexType>
```

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

Simple Type(s)

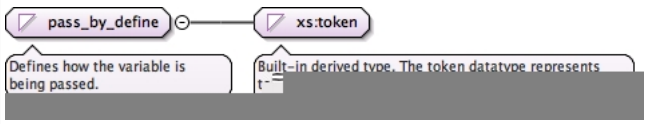
Simple Type `id_define`

Namespace	No namespace
Annotations	Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".
Diagram	
Type	restriction of xs:string
Facets	pattern ((0?x\d+)|\d+)
Used by	Attribute serializable/@typeid
Source	<xs:simpleType name="id_define"> <xs:annotation> <xs:documentation>Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:pattern value="((0?x\d+)|\d+)"/> </xs:restriction> </xs:simpleType>

Simple Type `full_items_define`

Namespace	No namespace
Annotations	Valid values for the full tag.
Diagram	
Type	restriction of xs:token
Facets	enumeration drop enumeration assert enumeration block
Source	<xs:simpleType name="full_items_define"> <xs:annotation> <xs:documentation>Valid values for the full tag.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="drop"/> <xs:enumeration value="assert"/> <xs:enumeration value="block"/> </xs:restriction> </xs:simpleType>

Simple Type `pass_by_define`

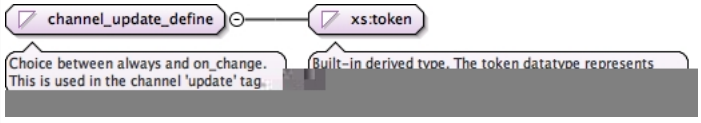
Namespace	No namespace
Annotations	Defines how the variable is being passed.
Diagram	
Type	restriction of xs:token
Facets	enumeration reference enumeration value enumeration pointer

Used by	Attributes arg_define/arg/@pass_by, return/@pass_by
Source	<xs:simpleType name="pass_by_define"> <xs:annotation> <xs:documentation>Defines how the variable is being passed.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="reference"/> <xs:enumeration value="value"/> <xs:enumeration value="pointer"/> </xs:restriction> </xs:simpleType>

Simple Type component_role_define

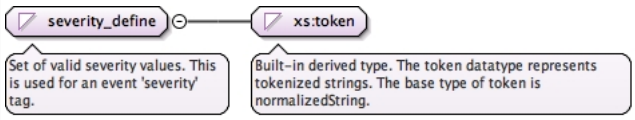
Namespace	No namespace
Annotations	Choice for component roles.
Diagram	
Type	restriction of xs:token
Facets	enumeration LogEvent enumeration LogTextEvent enumeration TimeGet enumeration ParamSet enumeration ParamGet enumeration Telemetry enumeration CmdRegistration enumeration Cmd enumeration CmdResponse
Source	<xs:simpleType name="component_role_define"> <xs:annotation> <xs:documentation>Choice for component roles.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="LogEvent"/> <xs:enumeration value="LogTextEvent"/> <xs:enumeration value="TimeGet"/> <xs:enumeration value="ParamSet"/> <xs:enumeration value="ParamGet"/> <xs:enumeration value="Telemetry"/> <xs:enumeration value="CmdRegistration"/> <xs:enumeration value="Cmd"/> <xs:enumeration value="CmdResponse"/> </xs:restriction> </xs:simpleType>

Simple Type channel_update_define

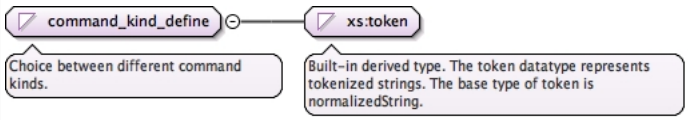
Namespace	No namespace
Annotations	Choice between always and on_change. This is used in the channel 'update' tag.
Diagram	
Type	restriction of xs:token
Facets	enumeration always enumeration on_change
Source	<xs:simpleType name="channel_update_define"> <xs:annotation> <xs:documentation>Choice between always and on_change. This is used in the channel 'update' tag.</xs:documentation> </xs:annotation>

```
  <xs:restriction base="xs:token">
    <xs:enumeration value="always"/>
    <xs:enumeration value="on_change"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type severity_define

Namespace	No namespace
Annotations	Set of valid severity values. This is used for an event 'severity' tag.
Diagram	
Type	restriction of xs:token
Facets	enumeration COMMAND enumeration ACTIVITY_LO enumeration ACTIVITY_HI enumeration WARNING_LO enumeration WARNING_HI enumeration DIAGNOSTIC enumeration FATAL
Source	<xs:simpleType name="severity_define"> <xs:annotation> <xs:documentation>Set of valid severity values. This is used for an event 'severity' tag.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="COMMAND"/> <xs:enumeration value="ACTIVITY_LO"/> <xs:enumeration value="ACTIVITY_HI"/> <xs:enumeration value="WARNING_LO"/> <xs:enumeration value="WARNING_HI"/> <xs:enumeration value="DIAGNOSTIC"/> <xs:enumeration value="FATAL"/> </xs:restriction> </xs:simpleType>

Simple Type command_kind_define

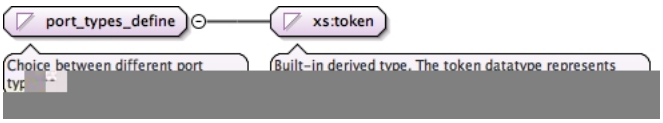
Namespace	No namespace
Annotations	Choice between different command kinds.
Diagram	
Type	restriction of xs:token
Facets	enumeration async enumeration sync enumeration guarded
Source	<xs:simpleType name="command_kind_define"> <xs:annotation> <xs:documentation>Choice between different command kinds.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="async"/> <xs:enumeration value="sync"/> <xs:enumeration value="guarded"/> </xs:restriction> </xs:simpleType>

Simple Type component_types_define

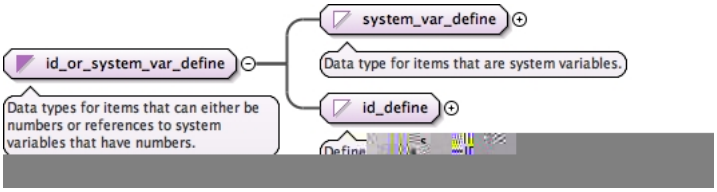
Namespace	No namespace

Annotations	Choice between active, passive, or queued.
Diagram	
Type	restriction of xs:token
Facets	enumeration active enumeration passive enumeration queued
Source	<xs:simpleType name="component_types_define"> <xs:annotation> <xs:documentation>Choice between active, passive, or queued.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="active"/> <xs:enumeration value="passive"/> <xs:enumeration value="queued"/> </xs:restriction> </xs:simpleType>

Simple Type port_types_define

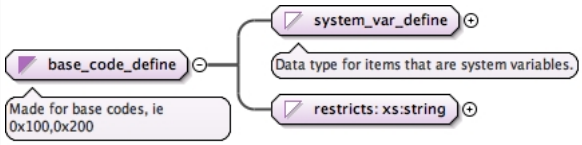
Namespace	No namespace
Annotations	Choice between different port types.
Diagram	
Type	restriction of xs:token
Facets	enumeration input enumeration sync_input enumeration guarded_input enumeration async_input enumeration model_input enumeration output
Source	<xs:simpleType name="port_types_define"> <xs:annotation> <xs:documentation>Choice between different port types.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="input"/> <xs:enumeration value="sync_input"/> <xs:enumeration value="guarded_input"/> <xs:enumeration value="async_input"/> <xs:enumeration value="model_input"/> <xs:enumeration value="output"/> </xs:restriction> </xs:simpleType>

Simple Type id_or_system_var_define

Namespace	No namespace
Annotations	Data types for items that can either be numbers or references to system variables that have numbers.
Diagram	
Type	union of(system_var_define, id_define)
Source	<xs:simpleType name="id_or_system_var_define">

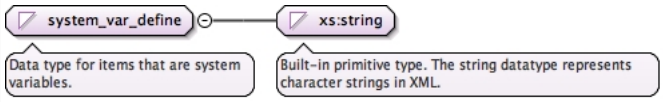
```
  <xs:annotation>
    <xs:documentation>Data types for items that can either be numbers or references to system
 variables that have numbers.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="system_var_define id_define"/>
</xs:simpleType>
```

Simple Type `base_code_define`

Namespace	No namespace
Annotations	`Made for base codes, ie 0x100,0x200`
Diagram	
Type	union of(system_var_define, restriction of xs:string)

Source	`<xs:simpleType name="base_code_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Made for base codes, ie 0x100,0x200</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:union memberTypes="system_var_define">`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:string">`
	`  <xs:pattern value="(((0?x\d+)|\d+)(,?))+"/>`
	`  </xs:restriction>`
	`  </xs:simpleType>`
	`  </xs:union>`
	`</xs:simpleType>`

Simple Type `system_var_define`

Namespace	No namespace
Annotations	`Data type for items that are system variables.`
Diagram	
Type	restriction of xs:string
Facets	pattern `$[\w|_|\-]+`

Source	`<xs:simpleType name="system_var_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Data type for items that are system variables.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:pattern value="$[\w|_|\-]+"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Attribute(s)

Attribute `item / @name`

Namespace	No namespace
Annotations	`Name of the enum item.`
Properties	use: `required`
Used by	Element item

Source	`<xs:attribute name="name" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation>Name of the enum item.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `item / @value`

Namespace	No namespace

Annotations	The value being sent through the enum item.
Used by	Element item
Source	<xs:attribute name="value"> <xs:annotation> <xs:documentation>The value being sent through the enum item.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute item / @comment

Namespace	No namespace
Annotations	Comment about the enum item.
Used by	Element item
Source	<xs:attribute name="comment"> <xs:annotation> <xs:documentation>Comment about the enum item.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute enum / @name

Namespace	No namespace
Annotations	Enum Name.
Properties	use: required
Used by	Element enum
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Enum Name.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute type_size_choice_define / @name

Namespace	No namespace
Annotations	Name of the memeber item.
Properties	use: required
Used by	Element member
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the memeber item.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute type_size_choice_define / @comment

Namespace	No namespace
Used by	Element member
Source	<xs:attribute name="comment"/>

Attribute type_size_choice_define / @format

Namespace	No namespace
Used by	Element member
Source	<xs:attribute name="format"/>

Attribute type_size_choice_define / @data_type

Namespace	No namespace
Type	union of(xs:string, restriction of xs:token, restriction of xs:token)
Properties	content: simple

Used by	Attribute Group type_size_choice_define
Source	<xs:attribute name="data_type"> <xs:simpleType> <xs:union memberTypes="xs:string"> <xs:simpleType> <xs:restriction base="xs:token"> <xs:enumeration value="string"/> </xs:restriction> </xs:simpleType> <xs:simpleType> <xs:restriction base="xs:token"> <xs:enumeration value="ENUM"/> </xs:restriction> </xs:simpleType> </xs:union> </xs:simpleType> </xs:attribute>

Attribute `type_size_choice_define / @type`

Namespace	No namespace
Type	union of(xs:string, restriction of xs:token, restriction of xs:token)
Properties	content: simple
Used by	Attribute Group type_size_choice_define
Source	<xs:attribute name="type"> <xs:simpleType> <xs:union memberTypes="xs:string"> <xs:simpleType> <xs:restriction base="xs:token"> <xs:enumeration value="string"/> </xs:restriction> </xs:simpleType> <xs:simpleType> <xs:restriction base="xs:token"> <xs:enumeration value="ENUM"/> </xs:restriction> </xs:simpleType> </xs:union> </xs:simpleType> </xs:attribute>

Attribute `type_size_choice_define / @size`

Namespace	No namespace
Annotations	The size of the argument.
Type	xs:nonNegativeInteger
Properties	content: simple
Used by	Attribute Group type_size_choice_define
Source	<xs:attribute name="size" type="xs:nonNegativeInteger"> <xs:annotation> <xs:documentation>The size of the argument.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `member / @comment`

Namespace	No namespace
Source	<xs:attribute name="comment"/>

Attribute `member / @format`

Namespace	No namespace
Source	<xs:attribute name="format"/>

Attribute `member / @name`

Namespace	No namespace

Annotations	Name of the memeber item.
Properties	use: required
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the memeber item.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute serializable / @name

Namespace	No namespace
Annotations	Name of the serializable object.
Properties	use: required
Used by	Element serializable
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the serializable object.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute serializable / @namespace

Namespace	No namespace
Annotations	Namespace of the serializable object.
Used by	Element serializable
Source	<xs:attribute name="namespace"> <xs:annotation> <xs:documentation>Namespace of the serializable object.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute serializable / @typeid

Namespace	No namespace
Annotations	ID for the object. If not declared, isfgen autogenerates one. Must be unique across all serlializable files.
Type	id_define
Properties	content: simple
Facets	pattern ((0?x\d+)|\d+)
Used by	Element serializable
Source	<xs:attribute name="typeid" type="id_define"> <xs:annotation> <xs:documentation>ID for the object. If not declared, isfgen autogenerates one. Must be unique across all serlializable files.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute arg_define / arg / @name

Namespace	No namespace
Annotations	Name of the argument.
Properties	use: required
Used by	Element arg_define/arg
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute arg_define / arg / @pass_by

Namespace	No namespace

Annotations	`Defines how the arguments are passed.`
Type	pass_by_define
Properties	content: `simple`
Facets	enumeration `reference` enumeration `value` enumeration `pointer`
Used by	Element arg_define/arg

Source	`<xs:attribute name="pass_by" type="pass_by_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines how the arguments are passed.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `arg_define / arg / @comment`

Namespace	No namespace
Annotations	`Comments about the argument.`
Type	xs:string
Properties	content: `simple`
Used by	Element arg_define/arg

Source	`<xs:attribute name="comment" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>Comments about the argument.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `return / @name`

Namespace	No namespace
Annotations	`Name of the argument.`
Used by	Element return

Source	`<xs:attribute name="name">`
	`  <xs:annotation>`
	`  <xs:documentation>Name of the argument.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `return / @pass_by`

Namespace	No namespace
Annotations	`Defines how the arguments are passed.`
Type	pass_by_define
Properties	content: `simple`
Facets	enumeration `reference` enumeration `value` enumeration `pointer`
Used by	Element return

Source	`<xs:attribute name="pass_by" type="pass_by_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines how the arguments are passed.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `return / @comment`

Namespace	No namespace
Annotations	`Comments about the argument.`
Type	xs:string

Properties	content:	simple
Used by	Element	return
Source	<xs:attribute name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Comments about the argument.</xs:documentation> </xs:annotation> </xs:attribute>	

Attribute external_arg_define / arg / @name

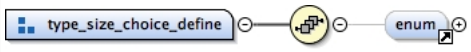
Namespace	No namespace
Annotations	Name of the argument.
Properties	use: required
Used by	Element external_arg_define/arg
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute external_arg_define / arg / @comment

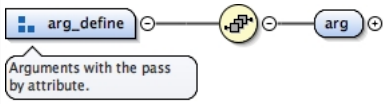
Namespace	No namespace
Annotations	Comments about the argument.
Type	xs:string
Properties	content: simple
Used by	Element external_arg_define/arg
Source	<xs:attribute name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Comments about the argument.</xs:documentation> </xs:annotation> </xs:attribute>

Element Group(s)

Element Group type_size_choice_define

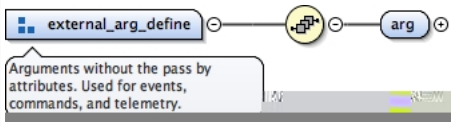
Namespace	No namespace
Diagram	
Used by	Elements arg_define/arg, external_arg_define/arg, member, return
Model	enum{0,1}
Children	enum
Source	<xs:group name="type_size_choice_define"> <xs:sequence> <xs:element minOccurs="0" ref="enum"/> </xs:sequence> </xs:group>

Element Group arg_define

Namespace	No namespace
Annotations	Arguments with the pass by attribute.
Diagram	
Model	arg
Children	arg

Source	<xs:group name="arg_define"> <xs:annotation> <xs:documentation>Arguments with the pass by attribute.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="arg"> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:group ref="type_size_choice_define"/> <xs:element ref="comment"/> </xs:choice> <xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="pass_by" type="pass_by_define"> <xs:annotation> <xs:documentation>Defines how the arguments are passed.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Comments about the argument.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attributeGroup ref="type_size_choice_define"/> </xs:complexType> </xs:element> </xs:sequence> </xs:group>

Element Group `external_arg_define`

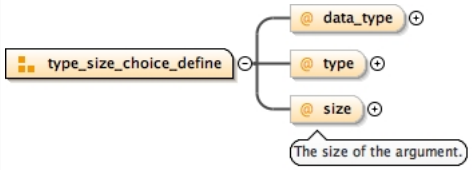
Namespace	No namespace
Annotations	Arguments without the pass by attributes. Used for events, commands, and telemetry.
Diagram	
Model	arg
Children	arg
Source	<xs:group name="external_arg_define"> <xs:annotation> <xs:documentation>Arguments without the pass by attributes. Used for events, commands, and telemetry.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="arg"> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:group ref="type_size_choice_define"/> <xs:element ref="comment"/> </xs:choice> <xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Comments about the argument.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attributeGroup ref="type_size_choice_define"/> </xs:complexType> </xs:element> </xs:sequence> </xs:group>

Attribute Group(s)

Attribute Group `type_size_choice_define`

Namespace	No namespace

Diagram



Used by

Elements arg_define/arg, external_arg_define/arg, member, return

Attributes

QName	Type	Use	
data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
size	xs:nonNegativeInteger	optional	
	`The size of the argument.`		
type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	

Source

```
<xs:attributeGroup name="type_size_choice_define">
  <xs:attribute name="data_type">
    <xs:simpleType>
      <xs:union memberTypes="xs:string">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="string"/>
          </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="ENUM"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="type">
    <xs:simpleType>
      <xs:union memberTypes="xs:string">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="string"/>
          </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="ENUM"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="size" type="xs:nonNegativeInteger">
    <xs:annotation>
      <xs:documentation>The size of the argument.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:attributeGroup>
```