Schema documentation for topology_schema.xsd

august 31, 2016

Table of Contents

Namespace: ""

Schema(s)

Main schema `topology_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`

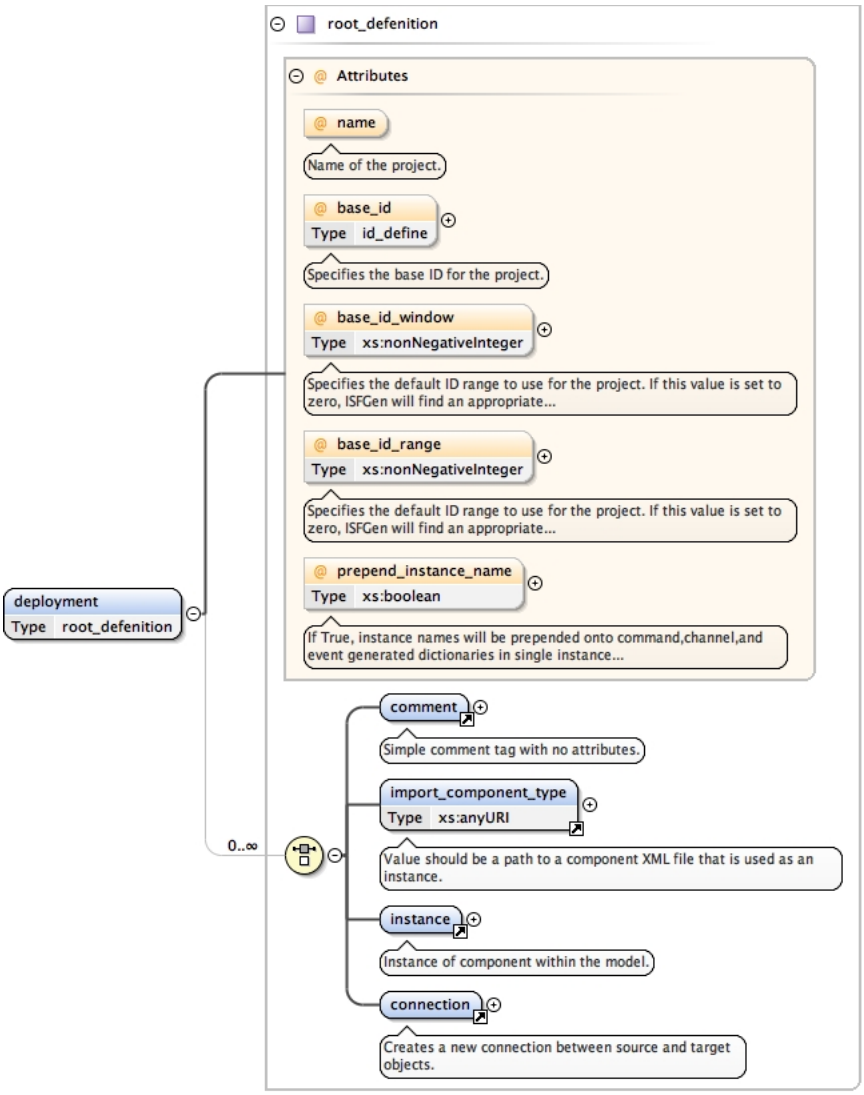
Included schema `common_types.xsd`

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

Element(s)

Element `deployment`

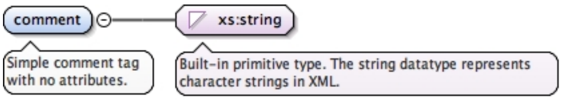
Namespace	No namespace

<table>
<tr><td>Diagram</td><td>
root_defenition
Attributes
name
Name of the project.
base_id
Type id_define
Specifies the base ID for the project.
base_id_window
Type xs:nonNegativeInteger
Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate...
base_id_range
Type xs:nonNegativeInteger
Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate...
prepend_instance_name
Type xs:boolean
If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance...
deployment
Type root_defenition
0..∞
comment
Simple comment tag with no attributes.
import_component_type
Type xs:anyURI
Value should be a path to a component XML file that is used as an instance.
instance
Instance of component within the model.
connection
Creates a new connection between source and target objects.

</td></tr>
<tr><td>Type</td><td>root_defenition</td></tr>
<tr><td>Properties</td><td>content: complex</td></tr>
<tr><td>Model</td><td>comment | import_component_type | instance | connection</td></tr>
<tr><td>Children</td><td>comment, connection, import_component_type, instance</td></tr>
<tr><td>Instance</td><td><deployment base_id="" base_id_range="" base_id_window="" name="" prepend_instance_name="">
  <comment>{1,1}</comment>
  <import_component_type>{1,1}</import_component_type>
  <instance base_id="" base_id_range="" base_id_window="" dict_short_name="" kind="" name="" namespace="" type="">
instance>
  <connection name="" type="">{1,1}</connection>
</deployment></td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr>
<tr><td><b>base_id</b></td><td>id_define</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Specifies the base ID for the project.</td></tr>
<tr><td><b>base_id_range</b></td><td>xs:nonNegativeInteger</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_window' can also be used.</td></tr>
<tr><td><b>base_id_window</b></td><td>xs:nonNegativeInteger</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_range' can also be used.</td></tr>
<tr><td><b>name</b></td><td></td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Name of the project.</td></tr>
</table>
</td></tr>
</table>

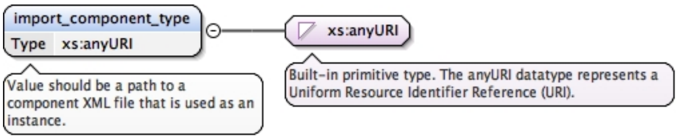
	QName	Type	Use	
	prepend_instance_name	xs:boolean	optional	
		If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance scenarios.		
Source	`<xs:element name="deployment" type="root_defenition"/>`			

Element comment

Namespace	No namespace
Annotations	Simple comment tag with no attributes.
Diagram	 comment xs:string Simple comment tag with no attributes. Built-in primitive type. The string datatype represents character strings in XML.
Type	xs:string
Properties	content: simple
Used by	Elements: arg_define/arg, connection, external_arg_define/arg, return Complex Type: root_defenition

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 import_component_type

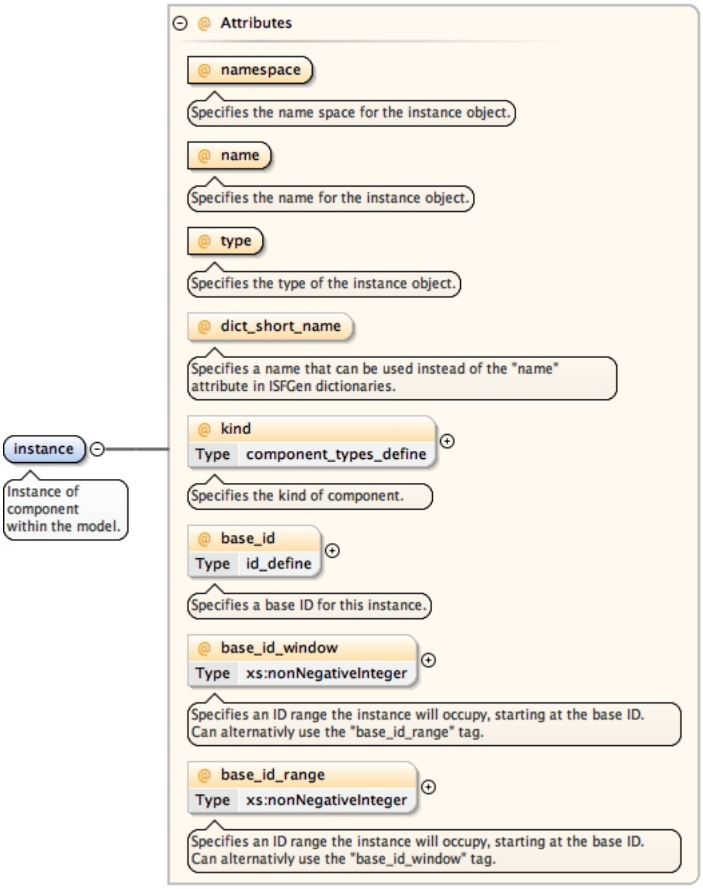
Namespace	No namespace
Annotations	Value should be a path to a component XML file that is used as an instance.
Diagram	 import_component_type Type xs:anyURI xs:anyURI Value should be a path to a component XML file that is used as an instance. Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
Type	xs:anyURI
Properties	content: simple
Used by	Complex Type: root_defenition

Source	`<xs:element name="import_component_type" type="xs:anyURI">`
	`  <xs:annotation>`
	`    <xs:documentation>Value should be a path to a component XML file that is used as an instance.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element instance

Namespace	No namespace
Annotations	Instance of component within the model.

Diagram


Attributes
namespace
Specifies the name space for the instance object.
name
Specifies the name for the instance object.
type
Specifies the type of the instance object.
dict_short_name
Specifies a name that can be used instead of the "name" attribute in ISFGen dictionaries.
instance
Instance of component within the model.
kind
Type component_types_define
Specifies the kind of component.
base_id
Type id_define
Specifies a base ID for this instance.
base_id_window
Type xs:nonNegativeInteger
Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_range" tag.
base_id_range
Type xs:nonNegativeInteger
Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_window" tag.

Properties	content:	`complex`
Used by	Complex Type	root_defenition

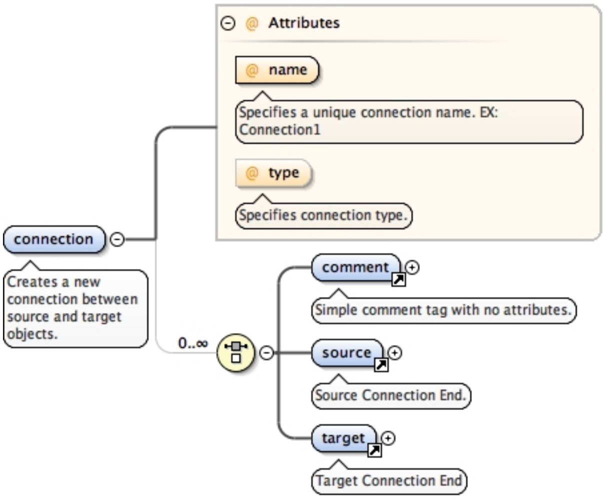
Attributes

QName	**Type**	**Use**	
base_id	id_define	optional	
	`Specifies a base ID for this instance.`		
base_id_range	xs:nonNegativeInteger	optional	
	`Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_window" tag.`		
base_id_window	xs:nonNegativeInteger	optional	
	`Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_range" tag.`		
dict_short_name		optional	
	`Specifies a name that can be used instead of the "name" attribute in ISFGen dictionaries.`		
kind	component_types_define	optional	
	`Specifies the kind of component.`		
name		required	
	`Specifies the name for the instance object.`		
namespace		required	
	`Specifies the name space for the instance object.`		
type		required	
	`Specifies the type of the instance object.`		

Source

```
<xs:element name="instance">
  <xs:annotation>
    <xs:documentation>Instance of component within the model.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="namespace" use="required">
      <xs:annotation>
    <xs:documentation>Specifies the name space for the instance object.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="name" use="required">
   <xs:annotation>
    <xs:documentation>Specifies the name for the instance object.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="type" use="required">
   <xs:annotation>
    <xs:documentation>Specifies the type of the instance object.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="dict_short_name">
   <xs:annotation>
    <xs:documentation>Specifies a name that can be used instead of the "name" attribute in
 ISFGen dictionaries.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="kind" type="component_types_define">
   <xs:annotation>
    <xs:documentation>Specifies the kind of component.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="base_id" type="id_define">
   <xs:annotation>
    <xs:documentation>Specifies a base ID for this instance.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="base_id_window" type="xs:nonNegativeInteger">
   <xs:annotation>
    <xs:documentation>Specifies an ID range the instance will occupy, starting at the base ID.
 Can alternativly use the "base_id_range" tag.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="base_id_range" type="xs:nonNegativeInteger">
   <xs:annotation>
    <xs:documentation>Specifies an ID range the instance will occupy, starting at the base ID.
 Can alternativly use the "base_id_window" tag.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

Element connection

Namespace	No namespace
Annotations	Creates a new connection between source and target objects.
Diagram	 Attributes name Specifies a unique connection name. EX: Connection1 type Specifies connection type. connection Creates a new connection between source and target objects. 0..∞ comment Simple comment tag with no attributes. source Source Connection End. target Target Connection End
Properties	content: complex
Used by	Complex Type root_defenition
Model	comment | source | target
Children	comment, source, target
Instance	`<connection name="" type="">`

```
  <comment>{1,1}</comment>
  <source component="" num="" port="" type="">{1,1}</source>
  <target component="" num="" port="" type="">{1,1}</target>
</connection>
```

Attributes

QName	Type	Use	
name		required	
	Specifies a unique connection name. EX: Connection1		
type		optional	
	Specifies connection type.		

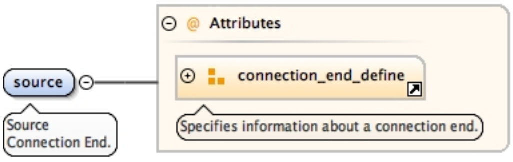
Source

```
<xs:element name="connection">
  <xs:annotation>
    <xs:documentation>Creates a new connection between source and target objects.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="comment"/>
      <xs:element ref="source"/>
      <xs:element ref="target"/>
    </xs:choice>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Specifies a unique connection name. EX: Connection1</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="type">
      <xs:annotation>
        <xs:documentation>Specifies connection type.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element source

Namespace	No namespace
Annotations	Source Connection End.

Diagram



Properties	content:	complex
Used by	Element	connection

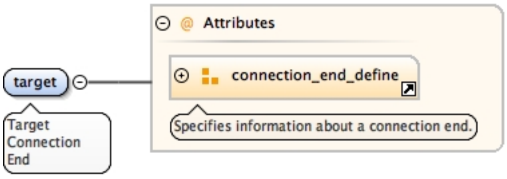
Attributes

QName	Type	Use	
component		required	
	Specifies the instance name of the connection end's component. Name must match an "name" attribute from specified "instance" tags.		
num	xs:nonNegativeInteger	optional	
	Specifies the multiplicity or index of a port that is being connected to. Generally, this value will be zero unless multiple indexes on the port exists.		
port		required	
	Specifies the port name on the object that the connection is attached to.		
type		required	
	Specifies the type of the connection end. Generally, this type will match the "target" type, unless connected to a port of type serial.		

Source

```
<xs:element name="source">
  <xs:annotation>
    <xs:documentation>Source Connection End.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attributeGroup ref="connection_end_define"/>
  </xs:complexType>
</xs:element>
```

Element `target`

Namespace	No namespace
Annotations	`Target Connection End`
Diagram	 Attributes connection_end_define Specifies information about a connection end. target Target Connection End
Properties	content: `complex`
Used by	Element connection

Attributes

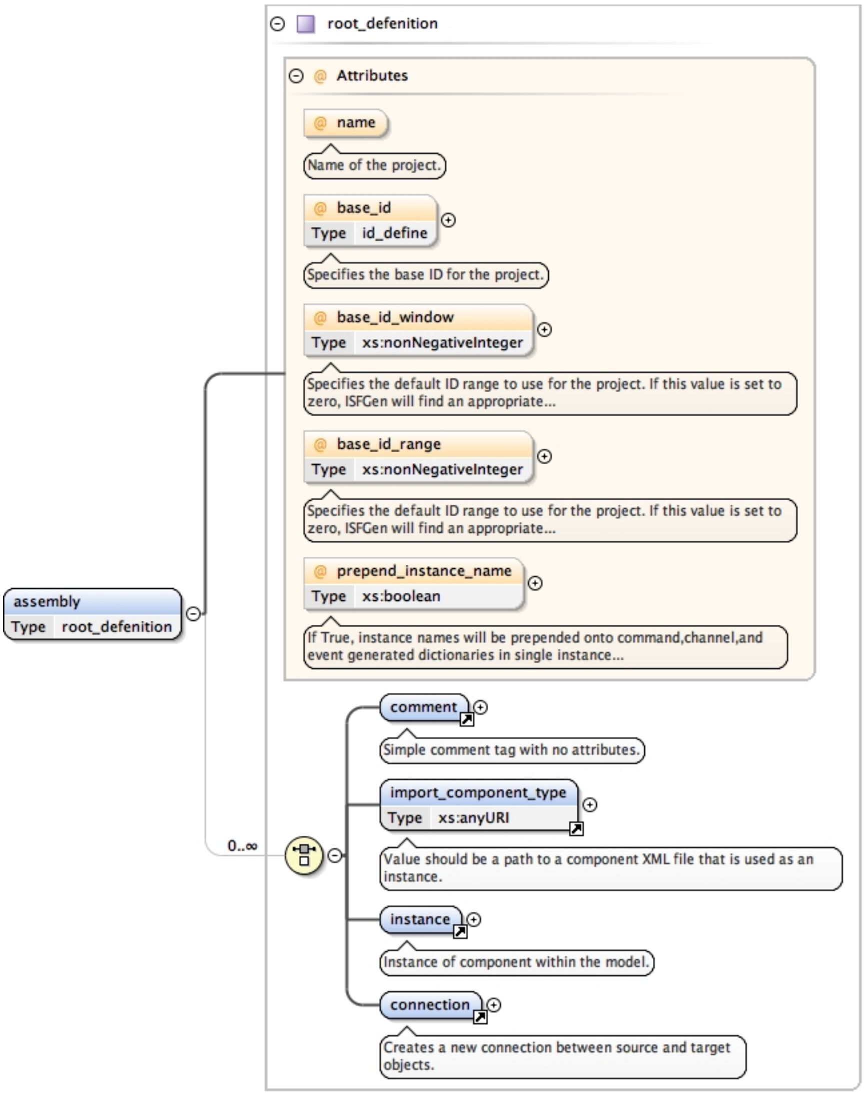
QName	Type	Use	
component		required	
	Specifies the instance name of the connection end's component. Name must match an "name" attribute from specified "instance" tags.		
num	xs:nonNegativeInteger	optional	
	Specifies the multiplicity or index of a port that is being connected to. Generally, this value will be zero unless multiple indexes on the port exists.		
port		required	
	Specifies the port name on the object that the connection is attached to.		
type		required	
	Specifies the type of the connection end. Generally, this type will match the "target" type, unless connected to a port of type serial.		

Source

```
<xs:element name="target">
  <xs:annotation>
    <xs:documentation>Target Connection End</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attributeGroup ref="connection_end_define"/>
  </xs:complexType>
</xs:element>
```

Element `assembly`

Namespace	No namespace

<table>
<tr><td>Diagram</td><td>
root_defenition
Attributes
name
Name of the project.
base_id
Type id_define
Specifies the base ID for the project.
base_id_window
Type xs:nonNegativeInteger
Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate...
base_id_range
Type xs:nonNegativeInteger
Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate...
prepend_instance_name
Type xs:boolean
If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance...
assembly
Type root_defenition
0..∞
comment
Simple comment tag with no attributes.
import_component_type
Type xs:anyURI
Value should be a path to a component XML file that is used as an instance.
instance
Instance of component within the model.
connection
Creates a new connection between source and target objects.

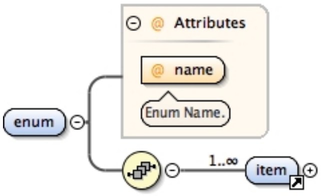
</td></tr>
<tr><td>Type</td><td>root_defenition</td></tr>
<tr><td>Properties</td><td>content: complex</td></tr>
<tr><td>Model</td><td>comment | import_component_type | instance | connection</td></tr>
<tr><td>Children</td><td>comment, connection, import_component_type, instance</td></tr>
<tr><td>Instance</td><td>

```
<assembly base_id="" base_id_range="" base_id_window="" name="" prepend_instance_name="">
  <comment>{1,1}</comment>
  <import_component_type>{1,1}</import_component_type>
  <instance base_id="" base_id_range="" base_id_window="" dict_short_name="" kind="" name="" namespace="" type="">
instance>
  <connection name="" type="">{1,1}</connection>
</assembly>
```

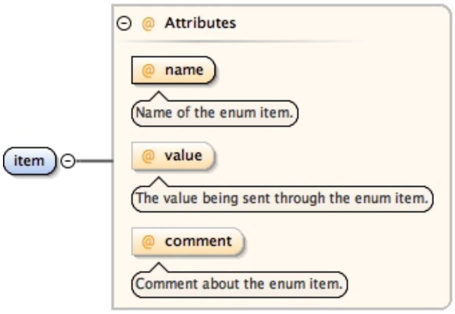
</td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr>
<tr><td><b>base_id</b></td><td>id_define</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Specifies the base ID for the project.</td></tr>
<tr><td><b>base_id_range</b></td><td>xs:nonNegativeInteger</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_window' can also be used.</td></tr>
<tr><td><b>base_id_window</b></td><td>xs:nonNegativeInteger</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_range' can also be used.</td></tr>
<tr><td><b>name</b></td><td></td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Name of the project.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr>
<tr><td><b>prepend_instance_name</b></td><td>xs:boolean</td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance scenarios.</td></tr>
</table></td></tr>
<tr><td>Source</td><td>&lt;xs:element name="assembly" type="root_defenition"/&gt;</td></tr>
</table>

Element enum

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Properties</td><td>content: complex</td></tr>
<tr><td>Used by</td><td>Element Group type_size_choice_define</td></tr>
<tr><td>Model</td><td>item+</td></tr>
<tr><td>Children</td><td>item</td></tr>
<tr><td>Instance</td><td>&lt;enum name="">
  <item comment="" name="" value="">{1,unbounded}</item>
</enum&gt;</td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr>
<tr><td><b>name</b></td><td></td><td>required</td><td></td></tr>
<tr><td></td><td colspan="3">Enum Name.</td></tr>
</table></td></tr>
<tr><td>Source</td><td>&lt;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&gt;</td></tr>
</table>

Element item

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
Attributes
name
Name of the enum item.
value
The value being sent through the enum item.
comment
Comment about the enum item.
item

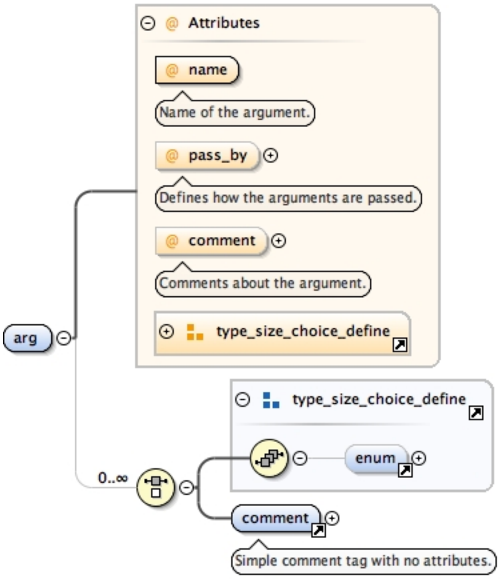
</td></tr>
<tr><td>Properties</td><td>content: complex</td></tr>
<tr><td>Used by</td><td>Element enum</td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr>
<tr><td><b>comment</b></td><td></td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3">Comment about the enum item.</td></tr>
</table></td></tr>
</table>

QName	Type	Use	
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 `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.		

QName	Type	Use	
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	

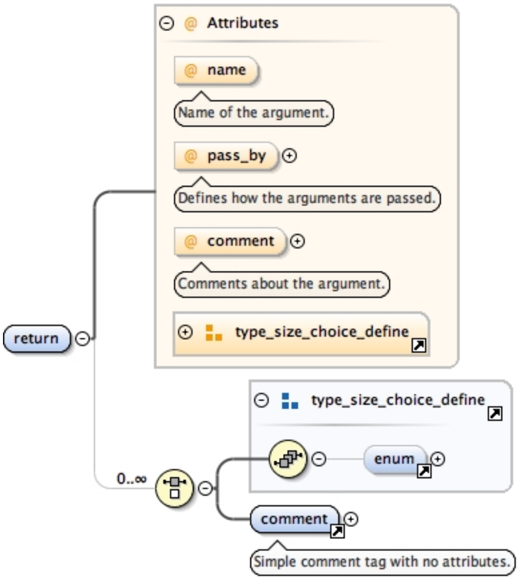
Source

```
<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

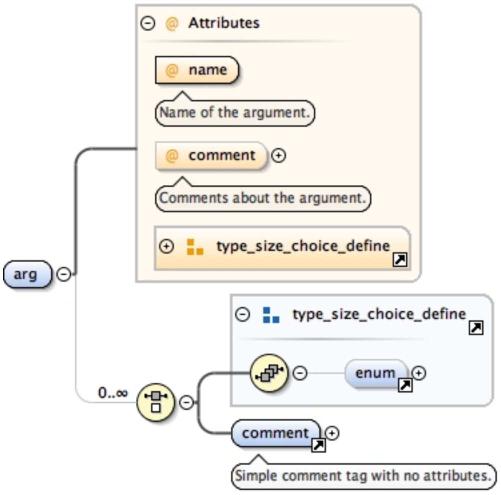
QName	Type	Use	
comment	xs:string	optional	

QName	Type	Use	
	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	 Attributes name Name of the argument. comment Comments about the argument. type_size_choice_define arg type_size_choice_define enum 0..∞ comment Simple comment tag with no attributes.
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

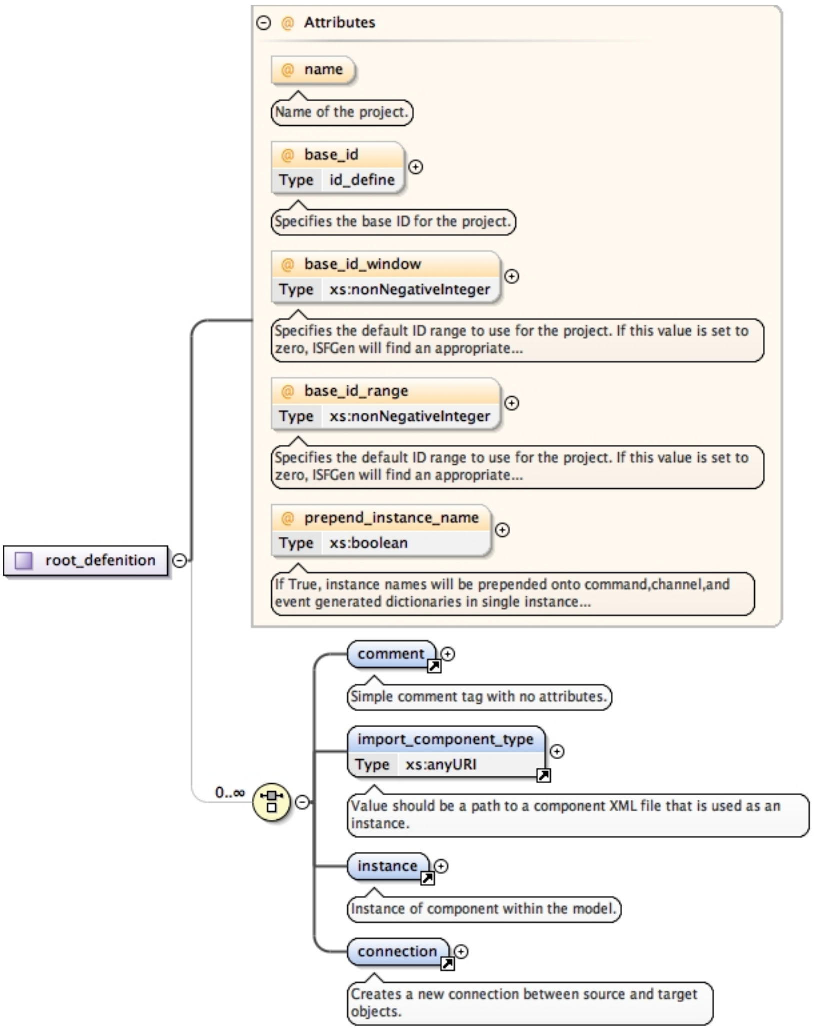
```
<xs:element name="arg">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="type_size_choice_define"/>
      <xs:element ref="comment"/>
    </xs:choice>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the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

Complex Type(s)

Complex Type `root_defenition`

Namespace	No namespace

Diagram


Attributes
name
Name of the project.
base_id
Type id_define
Specifies the base ID for the project.
base_id_window
Type xs:nonNegativeInteger
Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate...
base_id_range
Type xs:nonNegativeInteger
Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate...
prepend_instance_name
Type xs:boolean
If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance...
root_defenition
0..∞
comment
Simple comment tag with no attributes.
import_component_type
Type xs:anyURI
Value should be a path to a component XML file that is used as an instance.
instance
Instance of component within the model.
connection
Creates a new connection between source and target objects.

Used by	
Elements	assembly, deployment

Model: comment | import_component_type | instance | connection

Children: comment, connection, import_component_type, instance

Attributes

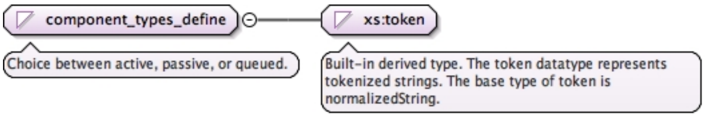
QName	Type	Use	
base_id	id_define	optional	
	Specifies the base ID for the project.		
base_id_range	xs:nonNegativeInteger	optional	
	Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_window' can also be used.		
base_id_window	xs:nonNegativeInteger	optional	
	Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_range' can also be used.		
name		optional	
	Name of the project.		
prepend_instance_name	xs:boolean	optional	
	If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance scenarios.		

Source

```
<xs:complexType name="root_defenition">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="comment"/>
    <xs:element ref="import_component_type"/>
    <xs:element ref="instance"/>
    <xs:element ref="connection"/>
  </xs:choice>
  <xs:attribute name="name">
    <xs:annotation>
      <xs:documentation>Name of the project.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="base_id" type="id_define">
    <xs:annotation>
      <xs:documentation>Specifies the base ID for the project.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="base_id_window" type="xs:nonNegativeInteger">
    <xs:annotation>
      <xs:documentation>Specifies the default ID range to use for the project. If this value is
set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag
'base_id_range' can also be used.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="base_id_range" type="xs:nonNegativeInteger">
    <xs:annotation>
      <xs:documentation>Specifies the default ID range to use for the project. If this value is
set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag
'base_id_window' can also be used.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="prepend_instance_name" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>If True, instance names will be prepended onto command,channel,and event
generated dictionaries in single instance scenarios.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

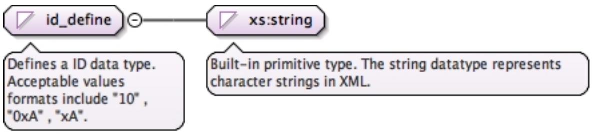
Simple Type(s)

Simple Type `component_types_define`

Namespace	No namespace
Annotations	Choice between active, passive, or queued.
Diagram	 component_types_define xs:token Choice between active, passive, or queued. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
Type	restriction of xs:token
Facets	enumeration active enumeration passive enumeration queued
Used by	Attribute instance/@kind

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 `id_define`

Namespace	No namespace
Annotations	Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".
Diagram	 id_define xs:string Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA". Built-in primitive type. The string datatype represents character strings in XML.
Type	restriction of xs:string

Facets	pattern ((0?x\d+)|\d+)
Used by	Attributes instance/@base_id, root_defenition/@base_id

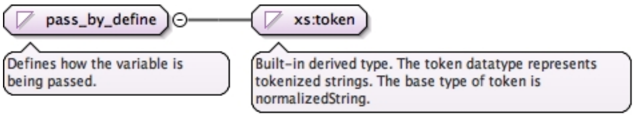
Source	<xs:simpleType name="id_define">
	  <xs:annotation>
	    <xs:documentation>Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".</xs:documentation>
	  </xs:annotation>
	  <xs:restriction base="xs:string">
	    <xs:pattern value="((0?x\d+)|\d+)"/>
	  </xs:restriction>
	</xs:simpleType>

Simple Type full_items_define

Namespace	No namespace
Annotations	Valid values for the full tag.
Diagram	full_items_define xs:token Valid values for the full tag. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
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>

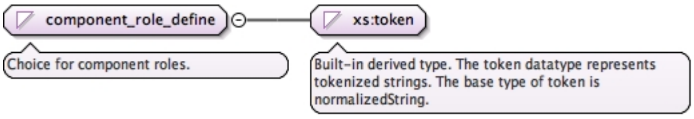
Simple Type pass_by_define

Namespace	No namespace
Annotations	Defines how the variable is being passed.
Diagram	 pass_by_define xs:token Defines how the variable is being passed. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
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

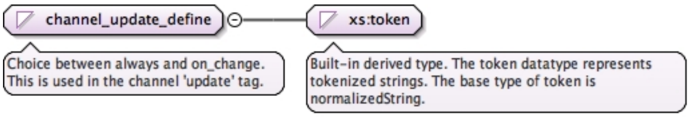
Namespace	No namespace
Annotations	Choice for component roles.

Diagram	 component_role_define xs:token Choice for component roles. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
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>
```

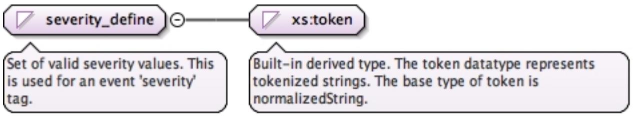
Simple Type channel_update_define

Namespace	No namespace
Annotations	Choice between always and on_change. This is used in the channel 'update' tag.
Diagram	 channel_update_define xs:token Choice between always and on_change. This is used in the channel 'update' tag. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
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

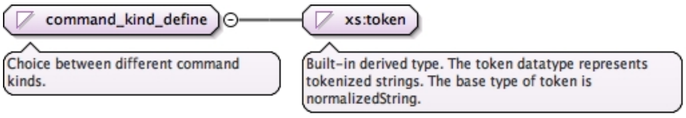
Namespace	No namespace
Annotations	Set of valid severity values. This is used for an event 'severity' tag.
Diagram	 severity_define xs:token Set of valid severity values. This is used for an event 'severity' tag. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
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	 command_kind_define xs:token Choice between different command kinds. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
Type	restriction of xs:token
Facets	enumeration async enumeration sync enumeration guarded

Source

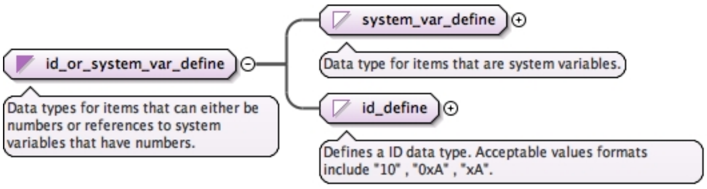
```
<xs:simpleType name="command_kind_define">
  <xs:annotation>
    <xs:documentation>Choice between different command kinds.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="async"/>
    <xs:enumeration value="sync"/>
    <xs:enumeration value="guarded"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `port_types_define`

Namespace	No namespace
Annotations	Choice between different port types.
Diagram	port_types_define xs:token Choice between different port types. Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.
Type	restriction of xs:token
Facets	enumeration input enumeration sync_input enumeration guarded_input enumeration async_input enumeration model_input enumeration output

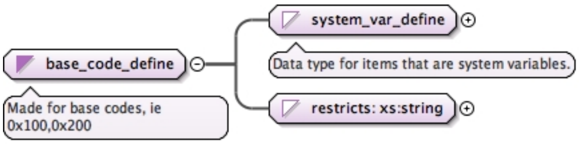
Source	`<xs:simpleType name="port_types_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Choice between different port types.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:token">`
	`    <xs:enumeration value="input"/>`
	`    <xs:enumeration value="sync_input"/>`
	`    <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	 id_or_system_var_define Data types for items that can either be numbers or references to system variables that have numbers. system_var_define Data type for items that are system variables. id_define Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".
Type	union of(system_var_define, id_define)

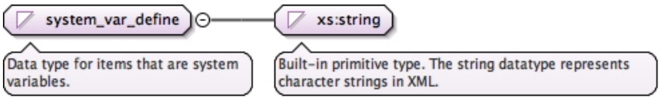
Source	`<xs:simpleType name="id_or_system_var_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Data types for items that can either be numbers or references to system variables that have numbers.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:union memberTypes="system_var_define id_define"/>`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

Simple Type base_code_define

Namespace	No namespace
Annotations	Made for base codes, ie 0x100,0x200
Diagram	 base_code_define Made for base codes, ie 0x100,0x200 system_var_define Data type for items that are system variables. restricts: xs:string
Type	union of(system_var_define, restriction of xs:string)

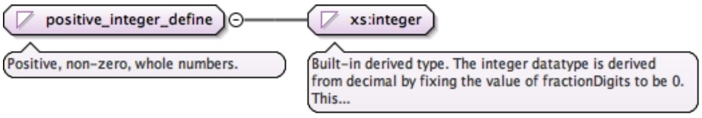
Source	`<xs:simpleType name="base_code_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Made for base codes, ie 0x100,0x200</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:union memberTypes="system_var_define">`
	`    <xs:simpleType>`
	`      <xs:restriction base="xs:string">`
	`        <xs:pattern 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	 system_var_define Data type for items that are system variables. xs:string Built-in primitive type. The string datatype represents character strings in XML.
Type	restriction of xs:string
Facets	pattern `$[\w|_|\-]+`

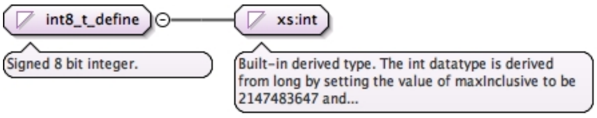
Source	`<xs:simpleType name="system_var_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Data type for items that are system variables.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`    <xs:pattern value="$[\w|_|\-]+"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type positive_integer_define

Namespace	No namespace
Annotations	Positive, non-zero, whole numbers.
Diagram	 positive_integer_define xs:integer Positive, non-zero, whole numbers. Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	restriction of xs:integer
Facets	minInclusive 1

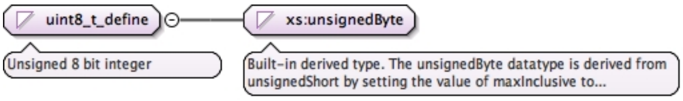
Source	`<xs:simpleType name="positive_integer_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Positive, non-zero, whole numbers.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:integer">`
	`    <xs:minInclusive value="1"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type int8_t_define

Namespace	No namespace
Annotations	Signed 8 bit integer.
Diagram	 int8_t_define xs:int Signed 8 bit integer. Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...
Type	restriction of xs:int
Facets	maxInclusive 127 minInclusive -128
Used by	Simple Type I8_define

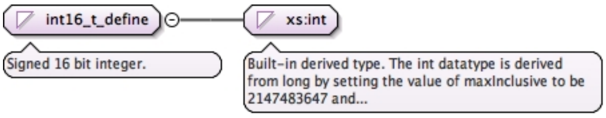
Source	`<xs:simpleType name="int8_t_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Signed 8 bit integer.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:int">`
	`    <xs:minInclusive value="-128"/>`
	`    <xs:maxInclusive value="127"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type uint8_t_define

Namespace	No namespace
Annotations	Unsigned 8 bit integer
Diagram	 uint8_t_define xs:unsignedByte Unsigned 8 bit integer Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...
Type	restriction of xs:unsignedByte
Facets	maxInclusive 255 minInclusive 0
Used by	Simple Type U8_define

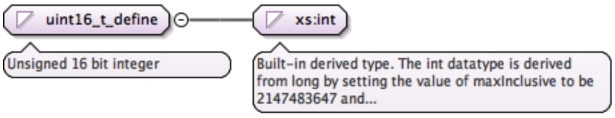
Source	`<xs:simpleType name="uint8_t_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Unsigned 8 bit integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:unsignedByte">`
	`    <xs:minInclusive value="0"/>`
	`    <xs:maxInclusive value="255"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `int16_t_define`

Namespace	No namespace
Annotations	`Signed 16 bit integer.`
Diagram	 int16_t_define xs:int Signed 16 bit integer. Built–in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...
Type	restriction of xs:int
Facets	maxInclusive `32767` minInclusive `-32768`
Used by	Simple Type I16_define

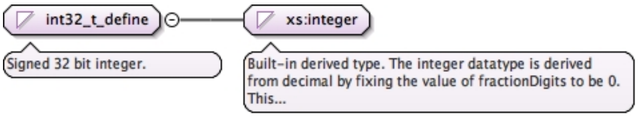
Source	`<xs:simpleType name="int16_t_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Signed 16 bit integer.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:int">`
	`    <xs:minInclusive value="-32768"/>`
	`    <xs:maxInclusive value="32767"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `uint16_t_define`

Namespace	No namespace
Annotations	`Unsigned 16 bit integer`
Diagram	 uint16_t_define xs:int Unsigned 16 bit integer Built–in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...
Type	restriction of xs:int
Facets	maxInclusive `65535` minInclusive `0`
Used by	Simple Type U16_define

Source	`<xs:simpleType name="uint16_t_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Unsigned 16 bit integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:int">`
	`    <xs:minInclusive value="0"/>`
	`    <xs:maxInclusive value="65535"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

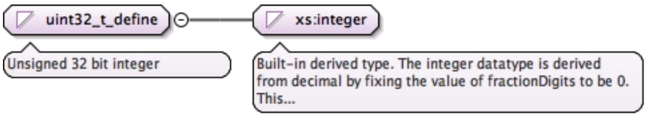
Simple Type `int32_t_define`

Namespace	No namespace
Annotations	`Signed 32 bit integer.`
Diagram	 int32_t_define xs:integer Signed 32 bit integer. Built–in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...

Type	restriction of xs:integer
Facets	maxInclusive 2147483647 minInclusive -2147483648
Used by	Simple Types I32_define, NATIVE_INT_TYPE_define

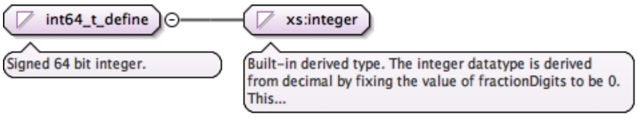
Source	<xs:simpleType name="int32_t_define">
	  <xs:annotation>
	    <xs:documentation>Signed 32 bit integer.</xs:documentation>
	  </xs:annotation>
	  <xs:restriction base="xs:integer">
	    <xs:minInclusive value="-2147483648"/>
	    <xs:maxInclusive value="2147483647"/>
	  </xs:restriction>
	</xs:simpleType>

Simple Type uint32_t_define

Namespace	No namespace
Annotations	Unsigned 32 bit integer
Diagram	 uint32_t_define xs:integer Unsigned 32 bit integer Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	restriction of xs:integer
Facets	maxInclusive 4294967295 minInclusive 0
Used by	Simple Types NATIVE_UINT_TYPE_define, U32_define

Source	<xs:simpleType name="uint32_t_define">
	  <xs:annotation>
	    <xs:documentation>Unsigned 32 bit integer</xs:documentation>
	  </xs:annotation>
	  <xs:restriction base="xs:integer">
	    <xs:minInclusive value="0"/>
	    <xs:maxInclusive value="4294967295"/>
	  </xs:restriction>
	</xs:simpleType>

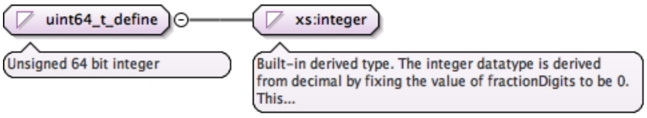
Simple Type int64_t_define

Namespace	No namespace
Annotations	Signed 64 bit integer.
Diagram	 int64_t_define xs:integer Signed 64 bit integer. Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	restriction of xs:integer
Facets	maxInclusive 9223372036854775807 minInclusive -9223372036854775808
Used by	Simple Type I64_define

Source	<xs:simpleType name="int64_t_define">
	  <xs:annotation>
	    <xs:documentation>Signed 64 bit integer.</xs:documentation>
	  </xs:annotation>
	  <xs:restriction base="xs:integer">
	    <xs:minInclusive value="-9223372036854775808"/>
	    <xs:maxInclusive value="9223372036854775807"/>
	  </xs:restriction>
	</xs:simpleType>

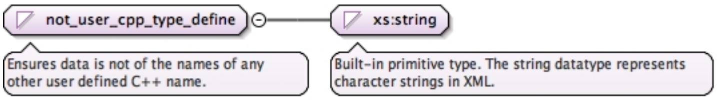
Simple Type uint64_t_define

Namespace	No namespace

Annotations	Unsigned 64 bit integer
Diagram	 uint64_t_define xs:integer Unsigned 64 bit integer Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	restriction of xs:integer
Facets	maxInclusive 18446744073709551615 minInclusive 0
Used by	Simple Type U64_define

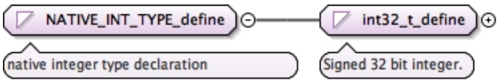
Source	`<xs:simpleType name="uint64_t_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Unsigned 64 bit integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:integer">`
	`    <xs:minInclusive value="0"/>`
	`    <xs:maxInclusive value="18446744073709551615"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type not_user_cpp_type_define

Namespace	No namespace
Annotations	Ensures data is not of the names of any other user defined C++ name.
Diagram	 not_user_cpp_type_define xs:string Ensures data is not of the names of any other user defined C++ name. Built-in primitive type. The string datatype represents character strings in XML.
Type	xs:string

Source	`<xs:simpleType name="not_user_cpp_type_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Ensures data is not of the names of any other user defined C++ name.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string"/>`
	`</xs:simpleType>`

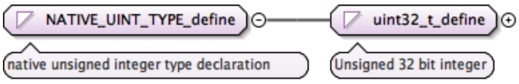
Simple Type NATIVE_INT_TYPE_define

Namespace	No namespace
Annotations	native integer type declaration
Diagram	 NATIVE_INT_TYPE_define int32_t_define native integer type declaration Signed 32 bit integer.
Type	int32_t_define
Type hierarchy	• xs:integer • int32_t_define • NATIVE_INT_TYPE_define
Facets	maxInclusive 2147483647 minInclusive -2147483648

Source	`<xs:simpleType name="NATIVE_INT_TYPE_define">`
	`  <xs:annotation>`
	`    <xs:documentation>native integer type declaration</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="int32_t_define"/>`
	`</xs:simpleType>`

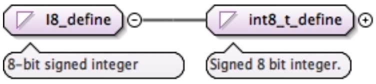
Simple Type NATIVE_UINT_TYPE_define

Namespace	No namespace
Annotations	native unsigned integer type declaration

Diagram	 NATIVE_UINT_TYPE_define native unsigned integer type declaration uint32_t_define Unsigned 32 bit integer
Type	uint32_t_define
Type hierarchy	- xs:integer - uint32_t_define - NATIVE_UINT_TYPE_define
Facets	maxInclusive 4294967295 minInclusive 0

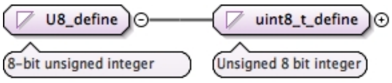
Source	`<xs:simpleType name="NATIVE_UINT_TYPE_define">`
	`  <xs:annotation>`
	`    <xs:documentation>native unsigned integer type declaration</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="uint32_t_define"/>`
	`</xs:simpleType>`

Simple Type I8_define

Namespace	No namespace
Annotations	8-bit signed integer
Diagram	 I8_define 8-bit signed integer int8_t_define Signed 8 bit integer.
Type	int8_t_define
Type hierarchy	- xs:int - int8_t_define - I8_define
Facets	maxInclusive 127 minInclusive -128

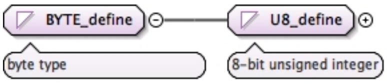
Source	`<xs:simpleType name="I8_define">`
	`  <xs:annotation>`
	`    <xs:documentation>8-bit signed integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="int8_t_define"/>`
	`</xs:simpleType>`

Simple Type U8_define

Namespace	No namespace
Annotations	8-bit unsigned integer
Diagram	 U8_define 8-bit unsigned integer uint8_t_define Unsigned 8 bit integer
Type	uint8_t_define
Type hierarchy	- xs:unsignedByte - uint8_t_define - U8_define
Facets	maxInclusive 255 minInclusive 0
Used by	Simple Type BYTE_define

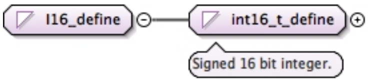
Source	`<xs:simpleType name="U8_define">`
	`  <xs:annotation>`
	`    <xs:documentation>8-bit unsigned integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="uint8_t_define"/>`
	`</xs:simpleType>`

Simple Type `BYTE_define`

Namespace	No namespace
Annotations	`byte type`
Diagram	 BYTE_define U8_define byte type 8-bit unsigned integer
Type	U8_define
Type hierarchy	• xs:unsignedByte • uint8_t_define • U8_define • BYTE_define
Facets	maxInclusive `255` minInclusive `0`

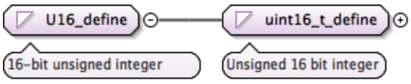
Source	`<xs:simpleType name="BYTE_define">`
	`  <xs:annotation>`
	`    <xs:documentation>byte type</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="U8_define"/>`
	`</xs:simpleType>`

Simple Type `I16_define`

Namespace	No namespace
Diagram	
Type	int16_t_define
Type hierarchy	• xs:int • int16_t_define • I16_define
Facets	maxInclusive `32767` minInclusive `-32768`

Source	`<xs:simpleType name="I16_define">`
	`  <xs:restriction base="int16_t_define"/>`
	`</xs:simpleType>`

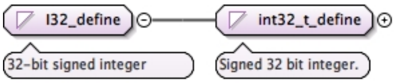
Simple Type `U16_define`

Namespace	No namespace
Annotations	`16-bit unsigned integer`
Diagram	 U16_define uint16_t_define 16-bit unsigned integer Unsigned 16 bit integer
Type	uint16_t_define
Type hierarchy	• xs:int • uint16_t_define • U16_define
Facets	maxInclusive `65535` minInclusive `0`

Source	`<xs:simpleType name="U16_define">`
	`  <xs:annotation>`
	`    <xs:documentation>16-bit unsigned integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="uint16_t_define"/>`

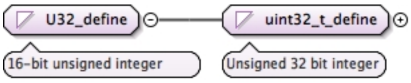
	`</xs:simpleType>`

Simple Type I32_define

Namespace	No namespace
Annotations	`32-bit signed integer`
Diagram	 I32_define int32_t_define 32-bit signed integer Signed 32 bit integer.
Type	int32_t_define
Type hierarchy	• xs:integer • int32_t_define • I32_define
Facets	maxInclusive `2147483647` minInclusive `-2147483648`

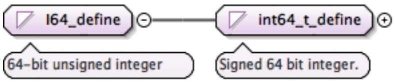
Source	`<xs:simpleType name="I32_define">`
	`  <xs:annotation>`
	`    <xs:documentation>32-bit signed integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="int32_t_define"/>`
	`</xs:simpleType>`

Simple Type U32_define

Namespace	No namespace
Annotations	`16-bit unsigned integer`
Diagram	 U32_define uint32_t_define 16-bit unsigned integer Unsigned 32 bit integer
Type	uint32_t_define
Type hierarchy	• xs:integer • uint32_t_define • U32_define
Facets	maxInclusive `4294967295` minInclusive `0`

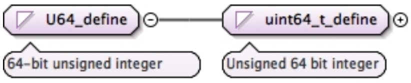
Source	`<xs:simpleType name="U32_define">`
	`  <xs:annotation>`
	`    <xs:documentation>16-bit unsigned integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="uint32_t_define"/>`
	`</xs:simpleType>`

Simple Type I64_define

Namespace	No namespace
Annotations	`64-bit unsigned integer`
Diagram	 I64_define int64_t_define 64-bit unsigned integer Signed 64 bit integer.
Type	int64_t_define
Type hierarchy	• xs:integer • int64_t_define • I64_define
Facets	maxInclusive `9223372036854775807` minInclusive `-9223372036854775808`
Source	`<xs:simpleType name="I64_define">`

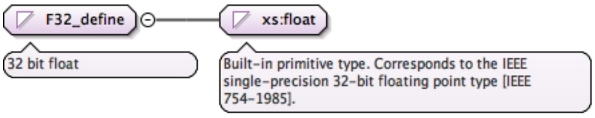
	`  <xs:annotation>`
	`    <xs:documentation>64-bit unsigned integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="int64_t_define"/>`
	`</xs:simpleType>`

Simple Type U64_define

Namespace	No namespace
Annotations	64-bit unsigned integer
Diagram	 U64_define uint64_t_define 64-bit unsigned integer Unsigned 64 bit integer
Type	uint64_t_define
Type hierarchy	• xs:integer • uint64_t_define • U64_define
Facets	maxInclusive 18446744073709551615 minInclusive 0

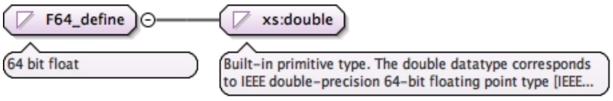
Source	`<xs:simpleType name="U64_define">`
	`  <xs:annotation>`
	`    <xs:documentation>64-bit unsigned integer</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="uint64_t_define"/>`
	`</xs:simpleType>`

Simple Type F32_define

Namespace	No namespace
Annotations	32 bit float
Diagram	 F32_define xs:float 32 bit float Built-in primitive type. Corresponds to the IEEE single-precision 32-bit floating point type [IEEE 754-1985].
Type	xs:float

Source	`<xs:simpleType name="F32_define">`
	`  <xs:annotation>`
	`    <xs:documentation>32 bit float</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:float"/>`
	`</xs:simpleType>`

Simple Type F64_define

Namespace	No namespace
Annotations	64 bit float
Diagram	 F64_define xs:double 64 bit float Built-in primitive type. The double datatype corresponds to IEEE double-precision 64-bit floating point type [IEEE...
Type	xs:double

Source	`<xs:simpleType name="F64_define">`
	`  <xs:annotation>`
	`    <xs:documentation>64 bit float</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:double"/>`
	`</xs:simpleType>`

Attribute(s)

Attribute instance / @namespace

Namespace	No namespace

Annotations	Specifies the name space for the instance object.
Properties	use: required
Used by	Element instance

Source	`<xs:attribute name="namespace" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the name space for the instance object.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute instance / @name

Namespace	No namespace
Annotations	Specifies the name for the instance object.
Properties	use: required
Used by	Element instance

Source	`<xs:attribute name="name" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the name for the instance object.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute instance / @type

Namespace	No namespace
Annotations	Specifies the type of the instance object.
Properties	use: required
Used by	Element instance

Source	`<xs:attribute name="type" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the type of the instance object.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute instance / @dict_short_name

Namespace	No namespace
Annotations	Specifies a name that can be used instead of the "name" attribute in ISFGen dictionaries.
Used by	Element instance

Source	`<xs:attribute name="dict_short_name">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies a name that can be used instead of the "name" attribute in ISFGen dictionaries.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute instance / @kind

Namespace	No namespace
Annotations	Specifies the kind of component.
Type	component_types_define
Properties	content: simple
Facets	enumeration active enumeration passive enumeration queued
Used by	Element instance

Source	`<xs:attribute name="kind" type="component_types_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the kind of component.</xs:documentation>`

	</xs:annotation> </xs:attribute>

Attribute instance / @base_id

Namespace	No namespace
Annotations	Specifies a base ID for this instance.
Type	id_define
Properties	content: simple
Facets	pattern ((0?x\d+)|\d+)
Used by	Element instance
Source	<xs:attribute name="base_id" type="id_define"> <xs:annotation> <xs:documentation>Specifies a base ID for this instance.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute instance / @base_id_window

Namespace	No namespace
Annotations	Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_range" tag.
Type	xs:nonNegativeInteger
Properties	content: simple
Used by	Element instance
Source	<xs:attribute name="base_id_window" type="xs:nonNegativeInteger"> <xs:annotation> <xs:documentation>Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_range" tag.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute instance / @base_id_range

Namespace	No namespace
Annotations	Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_window" tag.
Type	xs:nonNegativeInteger
Properties	content: simple
Used by	Element instance
Source	<xs:attribute name="base_id_range" type="xs:nonNegativeInteger"> <xs:annotation> <xs:documentation>Specifies an ID range the instance will occupy, starting at the base ID. Can alternativly use the "base_id_window" tag.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute connection_end_define / @component

Namespace	No namespace
Annotations	Specifies the instance name of the connection end's component. Name must match an "name" attribute from specified "instance" tags.
Properties	use: required
Used by	Attribute Group connection_end_define
Source	<xs:attribute name="component" use="required"> <xs:annotation> <xs:documentation>Specifies the instance name of the connection end's component. Name must match an "name" attribute from specified "instance" tags.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute connection_end_define / @port

Namespace	No namespace
Annotations	Specifies the port name on the object that the connection is attached to.
Properties	use: required
Used by	Attribute Group connection_end_define

Source	`<xs:attribute name="port" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the port name on the object that the connection is attached to.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute connection_end_define / @type

Namespace	No namespace
Annotations	Specifies the type of the connection end. Generally, this type will match the "target" type, unless connected to a port of type serial.
Properties	use: required
Used by	Attribute Group connection_end_define

Source	`<xs:attribute name="type" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the type of the connection end. Generally, this type will match the "target" type, unless connected to a port of type serial.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute connection_end_define / @num

Namespace	No namespace
Annotations	Specifies the multiplicity or index of a port that is being connected to. Generally, this value will be zero unless multiple indexes on the port exists.
Type	xs:nonNegativeInteger
Properties	content: simple
Used by	Attribute Group connection_end_define

Source	`<xs:attribute name="num" type="xs:nonNegativeInteger">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies the multiplicity or index of a port that is being connected to. Generally, this value will be zero unless multiple indexes on the port exists.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute connection / @name

Namespace	No namespace
Annotations	Specifies a unique connection name. EX: Connection1
Properties	use: required
Used by	Element connection

Source	`<xs:attribute name="name" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies a unique connection name. EX: Connection1</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute connection / @type

Namespace	No namespace
Annotations	Specifies connection type.
Used by	Element connection
Source	`<xs:attribute name="type">`

	<xs:annotation> <xs:documentation>Specifies connection type.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute root_defenition / @name

Namespace	No namespace	
Annotations	Name of the project.	
Used by	Complex Type	root_defenition
Source	<xs:attribute name="name"> <xs:annotation> <xs:documentation>Name of the project.</xs:documentation> </xs:annotation> </xs:attribute>	

Attribute root_defenition / @base_id

Namespace	No namespace	
Annotations	Specifies the base ID for the project.	
Type	id_define	
Properties	content:	simple
Facets	pattern	((0?x\d+)|\d+)
Used by	Complex Type	root_defenition
Source	<xs:attribute name="base_id" type="id_define"> <xs:annotation> <xs:documentation>Specifies the base ID for the project.</xs:documentation> </xs:annotation> </xs:attribute>	

Attribute root_defenition / @base_id_window

Namespace	No namespace	
Annotations	Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_range' can also be used.	
Type	xs:nonNegativeInteger	
Properties	content:	simple
Used by	Complex Type	root_defenition
Source	<xs:attribute name="base_id_window" type="xs:nonNegativeInteger"> <xs:annotation> <xs:documentation>Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_range' can also be used.</xs:documentation> </xs:annotation> </xs:attribute>	

Attribute root_defenition / @base_id_range

Namespace	No namespace	
Annotations	Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_window' can also be used.	
Type	xs:nonNegativeInteger	
Properties	content:	simple
Used by	Complex Type	root_defenition
Source	<xs:attribute name="base_id_range" type="xs:nonNegativeInteger"> <xs:annotation> <xs:documentation>Specifies the default ID range to use for the project. If this value is set to zero, ISFGen will find an appropriate range from the component XML files. Attribute tag 'base_id_window' can also be used.</xs:documentation> </xs:annotation>	

	</xs:attribute>

Attribute root_defenition / @prepend_instance_name

Namespace	No namespace
Annotations	If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance scenarios.
Type	xs:boolean
Properties	content: simple
Used by	Complex Type root_defenition
Source	<xs:attribute name="prepend_instance_name" type="xs:boolean"> <xs:annotation> <xs:documentation>If True, instance names will be prepended onto command,channel,and event generated dictionaries in single instance scenarios.</xs:documentation> </xs:annotation> </xs:attribute>

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 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 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 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`

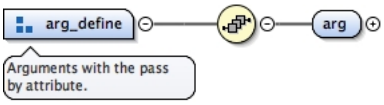
Namespace	No namespace
Annotations	Name of the argument.
Properties	use: required
Used by	Element external_arg_define/arg
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute>

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>

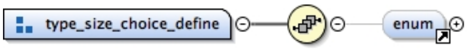
Element Group(s)

Element Group `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">

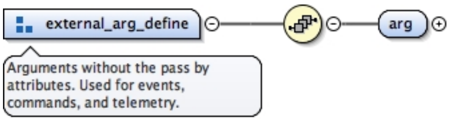
	<xs:group ref="type_size_choice_define"/>
	          <xs:element ref="comment"/>
	        </xs:choice>
	        <xs:attribute name="name" use="required">
	          <xs:annotation>
	            <xs:documentation>Name of the argument.</xs:documentation>
	          </xs:annotation>
	        </xs:attribute>
	        <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 type_size_choice_define

Namespace	No namespace
Diagram	
Used by	Elements arg_define/arg, external_arg_define/arg, 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 external_arg_define

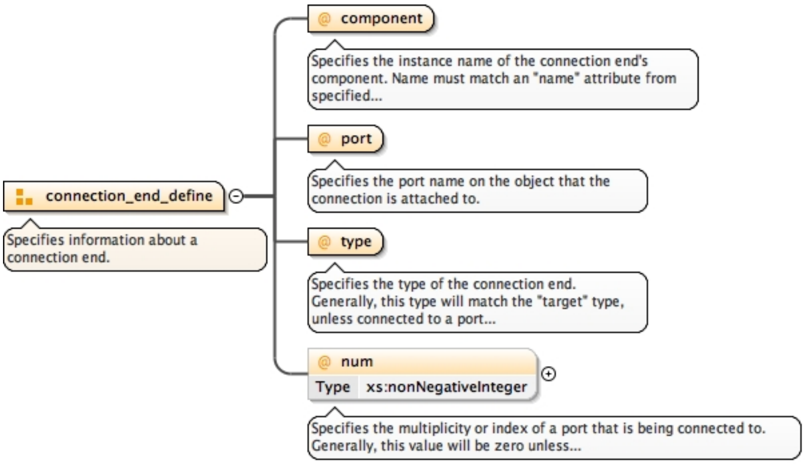
Namespace	No namespace
Annotations	Arguments without the pass by attributes. Used for events, commands, and telemetry.
Diagram	 external_arg_define arg Arguments without the pass by attributes. Used for events, commands, and telemetry.
Model	arg
Children	arg

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 connection_end_define

Namespace	No namespace
Annotations	Specifies information about a connection end.
Diagram	
Used by	Elements source, target

component
Specifies the instance name of the connection end's component. Name must match an "name" attribute from specified...
port
Specifies the port name on the object that the connection is attached to.
connection_end_define
Specifies information about a connection end.
type
Specifies the type of the connection end. Generally, this type will match the "target" type, unless connected to a port...
num
Type xs:nonNegativeInteger
Specifies the multiplicity or index of a port that is being connected to. Generally, this value will be zero unless...

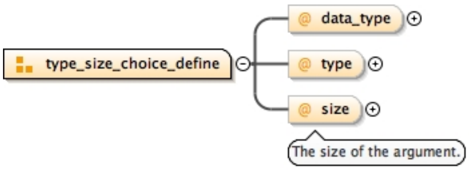
Attributes

QName	Type	Use	
component		required	
	Specifies the instance name of the connection end's component. Name must match an "name" attribute from specified "instance" tags.		
num	xs:nonNegativeInteger	optional	
	Specifies the multiplicity or index of a port that is being connected to. Generally, this value will be zero unless multiple indexes on the port exists.		
port		required	
	Specifies the port name on the object that the connection is attached to.		
type		required	
	Specifies the type of the connection end. Generally, this type will match the "target" type, unless connected to a port of type serial.		

Source

```
<xs:attributeGroup name="connection_end_define">
  <xs:annotation>
    <xs:documentation>Specifies information about a connection end.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="component" use="required">
    <xs:annotation>
      <xs:documentation>Specifies the instance name of the connection end's component. Name must
 match an "name" attribute from specified "instance" tags.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="port" use="required">
    <xs:annotation>
      <xs:documentation>Specifies the port name on the object that the connection is attached to.</
xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="type" use="required">
    <xs:annotation>
      <xs:documentation>Specifies the type of the connection end. Generally, this type will match
 the "target" type, unless connected to a port of type serial.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="num" type="xs:nonNegativeInteger">
    <xs:annotation>
      <xs:documentation>Specifies the multiplicity or index of a port that is being connected to.
 Generally, this value will be zero unless multiple indexes on the port exists.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:attributeGroup>
```

Attribute Group `type_size_choice_define`

Namespace	No namespace	
Diagram	 type_size_choice_define data_type type size The size of the argument.	
Used by	Elements	arg_define/arg, external_arg_define/arg, 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>
```