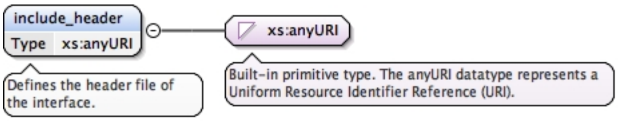
Schema documentation for interface_schema.xsd

august 31, 2016

Table of Contents

<table>
<tr><td>Properties</td><td><table><tr><td>content:</td><td><code>complex</code></td></tr><tr><td>abstract:</td><td><code>true</code></td></tr></table></td></tr>
<tr><td>Substitution Group</td><td>• interface
• port</td></tr>
<tr><td>Model</td><td>include_header | import_serializable_type | args | return | comment</td></tr>
<tr><td>Children</td><td>args, comment, import_serializable_type, include_header, return</td></tr>
<tr><td>Instance</td><td><code>&lt;interface_root_define name="" namespace="" priority="">
  <include_header>{1,1}</include_header>
  <import_serializable_type>{1,1}</import_serializable_type>
  <args>{1,1}</args>
  <return comment="" data_type="" name="" pass_by="" size="" type="">{1,1}</return>
  <comment>{1,1}</comment>
</interface_root_define&gt;</code></td></tr>
<tr><td>Attributes</td><td><table>
<tr><td>[illegible]</td><td>[illegible]</td><td>[illegible]</td><td></td></tr>
<tr><td>[illegible]</td><td></td><td>required</td><td></td></tr>
<tr><td></td><td colspan="3"><code>Interface name.</code></td></tr>
<tr><td>[illegible]</td><td></td><td>optional</td><td></td></tr>
<tr><td></td><td colspan="3"><code>Namespace in which the interface is in.</code></td></tr>
<tr><td>[illegible]</td><td>xs:integer</td><td>optional</td><td></td></tr>
</table></td></tr>
<tr><td>Source</td><td><code>&lt;xs:element name="interface_root_define" abstract="true" type="interface_define">
  <xs:annotation>
    <xs:documentation>Used for interface files.</xs:documentation>
  </xs:annotation>
</xs:element&gt;</code></td></tr>
</table>

Element `include_header`

Namespace	No namespace
Annotations	`Defines the header file of the interface.`
Diagram	 include_header Type xs:anyURI xs:anyURI Defines the header file of the interface. Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
Type	xs:anyURI
Properties	content: `simple`
Used by	Complex Type interface_define

Source	`<xs:element name="include_header" type="xs:anyURI">`
	`  <xs:annotation>`
	`    <xs:documentation>Defines the header file of the interface.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

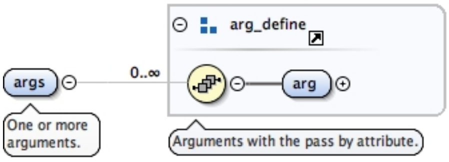
Element `import_serializable_type`

Namespace	No namespace
Annotations	`Imports serlializable types.`
Diagram	
Type	xs:anyURI
Properties	content: `simple`
Used by	Complex Type interface_define

Source	`<xs:element name="import_serializable_type" type="xs:anyURI">`
	`  <xs:annotation>`
	`    <xs:documentation>Imports serlializable types.</xs:documentation>`
	`  </xs:annotation>`

	`</xs:element>`

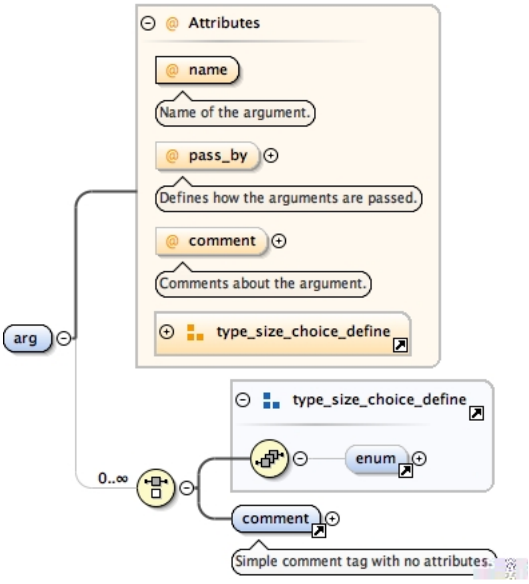
Element `args`

Namespace	No namespace
Annotations	`One or more arguments.`
Diagram	 arg_define args 0..∞ arg One or more arguments. Arguments with the pass by attribute.
Properties	content: `complex`
Used by	Complex Type interface_define
Model	arg
Children	arg

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

Source	`<xs:element name="args">`
	`<xs:annotation>`
	`<xs:documentation>One or more arguments.</xs:documentation>`
	`</xs:annotation>`
	`<xs:complexType>`
	`<xs:group minOccurs="0" maxOccurs="unbounded" ref="arg_define"/>`
	`</xs:complexType>`
	`</xs:element>`

Element `arg_define / arg`

Namespace	No namespace
Diagram	 Attributes name Name of the argument. pass_by Defines how the arguments are passed. comment Comments about the argument. type_size_choice_define arg type_size_choice_define 0..∞ enum comment Simple comment tag with no attributes.
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	

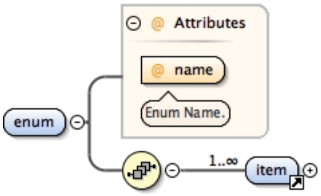
[illegible]	[illegible]	[illegible]	
[illegible]	xs:string	optional	

e Nn	a Q e	y T p e U	s	
		Comments about the argument.		
t te	a a _ y d p	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
e m	a n		required	
		Name of the argument.		
	a _ s s y p b	pass_by_define	optional	
		Defines how the arguments are passed.		
z e	s i	xs:nonNegativeInteger	optional	
		The size of the argument.		
t e	y p	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 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

e Nn	a Q e	y T p e U	s	
e m	a n		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

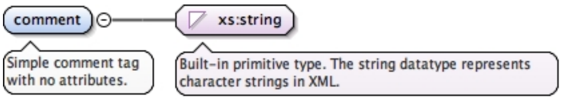
Attributes

[illegible]	[illegible]	[illegible]	
[illegible]		optional	
	Comment about the enum item.		
[illegible]		required	
	Name of the enum item.		
[illegible]		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>
```

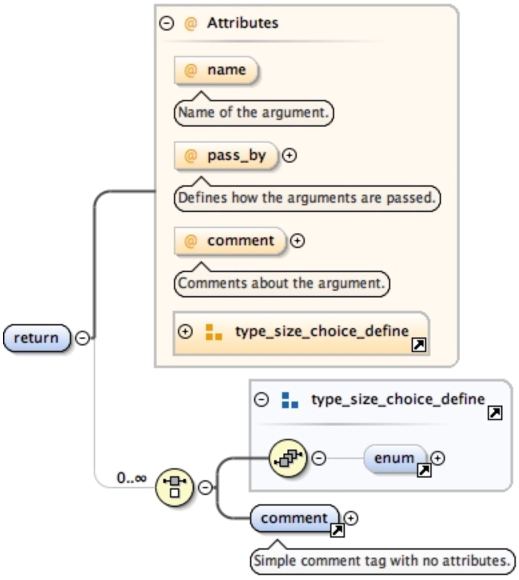
Element comment

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, external_arg_define/arg, return
	Complex Type	interface_define

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 `return`

Namespace	No namespace
Diagram	
Properties	content: complex
Used by	Complex Type interface_define
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

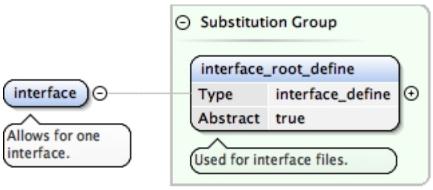
e Nn	a Q e	y T p	e U	
c m e m	t o n	xs:string	optional	
		Comments about the argument.		
t te	a a _ y d p	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
e m	a n		optional	
		Name of the argument.		
	a _ s s y p b	pass_by_define	optional	
		Defines how the arguments are passed.		
z e	s i	xs:nonNegativeInteger	optional	
		The size of the argument.		
t e	y p	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 interface

Namespace	No namespace
Annotations	Allows for one interface.
Diagram	 Substitution Group interface_root_define Type interface_define Abstract true Used for interface files. interface Allows for one interface.
Substitution Group Affiliation	- interface_root_define

Source	`<xs:element name="interface" substitutionGroup="interface_root_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Allows for one interface.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

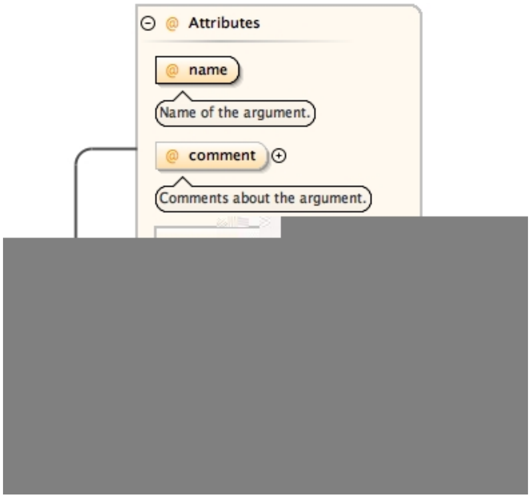
Element port

Namespace	No namespace
Annotations	Allows for one interface.
Diagram	
Substitution Group Affiliation	- interface_root_define

Source	`<xs:element name="port" substitutionGroup="interface_root_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Allows for one interface.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element external_arg_define / arg

Namespace	No namespace

Diagram	 Attributes name Name of the argument. comment Comments about the argument. [redacted]
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 e Nn	a Q e	y T p e U		
c m e m	t o n	xs:string	optional	
		Comments about the argument.		
t te	a a _ y d p	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
e m	a n		required	
		Name of the argument.		
z e	s i	xs:nonNegativeInteger	optional	
		The size of the argument.		
t e	y p	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	

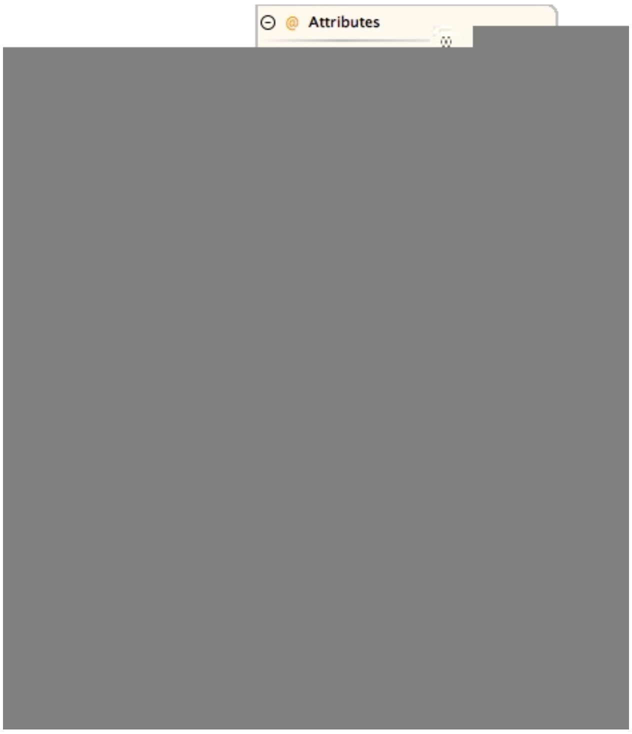
Source

```
<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 `interface_define`

Namespace	No namespace

Diagram	
Used by	Element interface_root_define
Model	include_header | import_serializable_type | args | return | comment
Children	args, comment, import_serializable_type, include_header, return

Attributes

e Nn	a Q e	y T p e U	s	
e m	a n		required	
		Interface name.		
e m c	e a as n p		optional	
		Namespace in which the interface is in.		
r r t	o y i ip	xs:integer	optional	

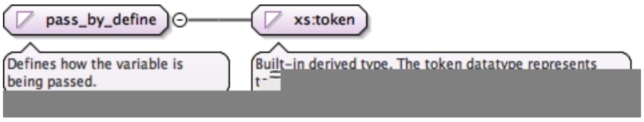
Source

```
<xs:complexType name="interface_define">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="include_header"/>
    <xs:element ref="import_serializable_type"/>
    <xs:element ref="args"/>
    <xs:element ref="return"/>
    <xs:element ref="comment"/>
  </xs:choice>
  <xs:attribute name="name" use="required">
    <xs:annotation>
      <xs:documentation>Interface name.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="priority" type="xs:integer"/>
  <xs:attribute name="namespace">
    <xs:annotation>
      <xs:documentation>Namespace in which the interface is in.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Simple Type(s)

Simple Type pass_by_define

Namespace	No namespace
Annotations	Defines how the variable is being passed.

<table>
<tr><td>Diagram</td><td>

pass_by_define
xs:token
Defines how the variable is being passed.
Built-in derived type. The token datatype represents

</td></tr>
<tr><td>Type</td><td>restriction of xs:token</td></tr>
<tr><td>Facets</td><td>enumeration reference
enumeration value
enumeration pointer</td></tr>
<tr><td>Used by</td><td>Attributes arg_define/arg/@pass_by, return/@pass_by</td></tr>
<tr><td>Source</td><td><pre>
<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>
</pre></td></tr>
</table>

Simple Type `full_items_define`

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td>Valid values for the full tag.</td></tr>
<tr><td>Diagram</td><td>

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.

</td></tr>
<tr><td>Type</td><td>restriction of xs:token</td></tr>
<tr><td>Facets</td><td>enumeration drop
enumeration assert
enumeration block</td></tr>
<tr><td>Source</td><td><pre>
<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>
</pre></td></tr>
</table>

Simple Type `component_role_define`

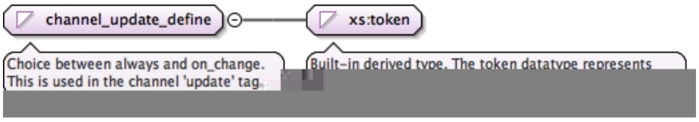
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td>Choice for component roles.</td></tr>
<tr><td>Diagram</td><td></td></tr>
<tr><td>Type</td><td>restriction of xs:token</td></tr>
<tr><td>Facets</td><td>enumeration LogEvent
enumeration LogTextEvent
enumeration TimeGet
enumeration ParamSet
enumeration ParamGet
enumeration Telemetry
enumeration CmdRegistration</td></tr>
</table>

	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

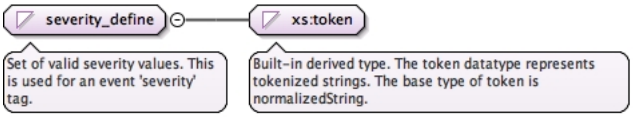
Namespace	No namespace
Annotations	Choice between always and on_change. This is used in the channel 'update' tag.
Diagram	
Type	restriction of xs:token
Facets	enumeration always enumeration on_change

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

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

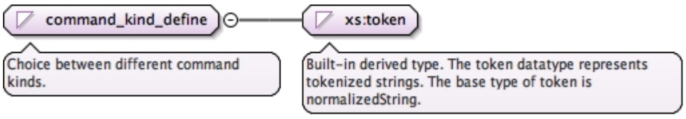
Namespace	No namespace
Annotations	Set of valid severity values. This is used for an event 'severity' tag.
Diagram	
Type	restriction of xs:token
Facets	enumeration COMMAND enumeration ACTIVITY_LO enumeration ACTIVITY_HI enumeration WARNING_LO enumeration WARNING_HI enumeration DIAGNOSTIC enumeration FATAL

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.

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 component_types_define

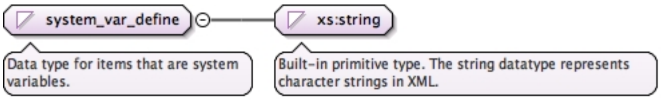
Namespace	No namespace
Annotations	Choice between active, passive, or queued.
Diagram	
Type	restriction of xs:token
Facets	enumeration active enumeration passive enumeration queued

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

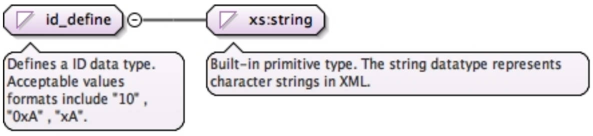
Namespace	No namespace
Annotations	Choice between different port types.
Diagram	port_types_define xs:token Choice between different port typ Built-in derived type. The token datatype represents
Type	restriction of xs:token

<table>
<tr><td>Diagram</td><td>

system_var_define
xs:string
Data type for items that are system variables.
Built-in primitive type. The string datatype represents character strings in XML.

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

Simple Type id_define

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td>Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".</td></tr>
<tr><td>Diagram</td><td>

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.

</td></tr>
<tr><td>Type</td><td>restriction of xs:string</td></tr>
<tr><td>Facets</td><td>pattern   ((0?x\d+)|\d+)</td></tr>
<tr><td>Source</td><td><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></td></tr>
</table>

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 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 interface_define / @name

Namespace	No namespace
Annotations	Interface name.
Properties	use: required
Used by	Complex Type interface_define

Source	`<xs:attribute name="name" use="required">`
	`  <xs:annotation>`
	`    <xs:documentation>Interface name.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute interface_define / @priority

Namespace	No namespace
Type	xs:integer
Properties	content: simple
Used by	Complex Type interface_define
Source	`<xs:attribute name="priority" type="xs:integer"/>`

Attribute interface_define / @namespace

Namespace	No namespace
Annotations	Namespace in which the interface is in.

Used by	Complex Type interface_define

Source	`<xs:attribute name="namespace">`
	`  <xs:annotation>`
	`    <xs:documentation>Namespace in which the interface is in.</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>`

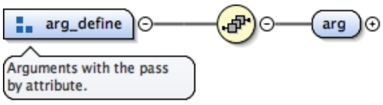
Attribute external_arg_define / arg / @comment

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)

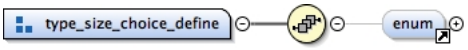
Element Group arg_define

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

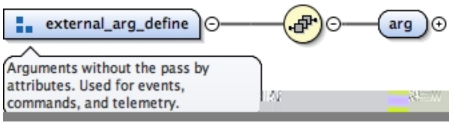
Source	`<xs:group name="arg_define">`
	`  <xs:annotation>`
	`    <xs:documentation>Arguments with the pass by attribute.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`    <xs:element name="arg">`
	`      <xs:complexType>`
	`        <xs:choice minOccurs="0" maxOccurs="unbounded">`
	`          <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>`

<table>
<tr><td></td><td><pre>
          </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>
</pre></td></tr>
</table>

Element Group `type_size_choice_define`

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

Element Group `external_arg_define`

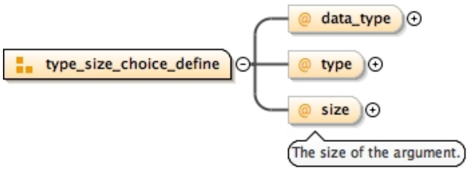
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td>Arguments without the pass by attributes. Used for events, commands, and telemetry.</td></tr>
<tr><td>Diagram</td><td>
external_arg_define
arg
Arguments without the pass by attributes. Used for events, commands, and telemetry.

</td></tr>
<tr><td>Model</td><td>arg</td></tr>
<tr><td>Children</td><td>arg</td></tr>
<tr><td>Source</td><td><pre>
<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>
</pre></td></tr>
</table>

Attribute Group(s)

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 e Na	a Q e	y T p e U	s	
t te	a a _ y d p	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
z e	s i	xs:nonNegativeInteger	optional	
		The size of the argument.		
te	y p	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>
```