

# Schema documentation for parameters\_schema.xsd

august 31, 2016

## Table of Contents

Namespace: ""	2
Schema(s)	2
Main schema parameters_schema.xsd	2
Included schema common_elements.xsd	2
Included schema common_types.xsd	2
Element(s)	2
Element parameters	2
Element parameter	3
Element enum	4
Element item	5
Element comment	5
Element arg_define / arg	6
Element return	7
Element external_arg_define / arg	8
Complex Type(s)	9
Complex Type data_type_and_default_define	9
Simple Type(s)	11
Simple Type positive_integer_define	11
Simple Type id_define	11
Simple Type base_code_define	11
Simple Type full_items_define	12
Simple Type pass_by_define	12
Simple Type component_role_define	12
Simple Type channel_update_define	13
Simple Type severity_define	13
Simple Type command_kind_define	14
Simple Type component_types_define	14
Simple Type port_types_define	14
Simple Type id_or_system_var_define	15
Simple Type system_var_define	15
Simple Type int8_t_define	16
Simple Type uint8_t_define	16
Simple Type int16_t_define	16
Simple Type uint16_t_define	17
Simple Type int32_t_define	17
Simple Type uint32_t_define	17
Simple Type int64_t_define	18
Simple Type uint64_t_define	18
Simple Type not_user_cpp_type_define	18
Simple Type NATIVE_INT_TYPE_define	19
Simple Type NATIVE_UINT_TYPE_define	19
Simple Type I8_define	19
Simple Type U8_define	20
Simple Type BYTE_define	20
Simple Type I16_define	20
Simple Type U16_define	21
Simple Type I32_define	21
Simple Type U32_define	21
Simple Type I64_define	22
Simple Type U64_define	22
Simple Type F32_define	23
Simple Type F64_define	23
Attribute(s)	23
Attribute item / @name	23
Attribute item / @value	23
Attribute item / @comment	23
Attribute enum / @name	24
Attribute data_type_and_default_define / @data_type	24
Attribute data_type_and_default_define / @default	25
Attribute data_type_and_default_define / @size	25

Attribute parameter / @id .....	25
Attribute parameter / @set_opcode .....	25
Attribute parameter / @save_opcode .....	26
Attribute parameter / @name .....	26
Attribute parameters / @parameter_base .....	26
Attribute parameters / @opcode_base .....	26
Attribute arg_define / arg / @name .....	27
Attribute arg_define / arg / @pass_by .....	27
Attribute arg_define / arg / @comment .....	27
Attribute type_size_choice_define / @data_type .....	27
Attribute type_size_choice_define / @type .....	28
Attribute type_size_choice_define / @size .....	28
Attribute return / @name .....	28
Attribute return / @pass_by .....	28
Attribute return / @comment .....	29
Attribute external_arg_define / arg / @name .....	29
Attribute external_arg_define / arg / @comment .....	29
Element Group(s) .....	29
Element Group arg_define .....	29
Element Group type_size_choice_define .....	30
Element Group external_arg_define .....	30
Attribute Group(s) .....	31
Attribute Group type_size_choice_define .....	31

## Namespace: ""

## Schema(s)

### Main schema parameters\_schema.xsd

Namespace	No namespace
Properties	<div>attribute form default: unqualified</div> <div>element form default: qualified</div>

### Included schema common\_elements.xsd

Namespace	No namespace
Properties	<div>attribute form default: unqualified</div> <div>element form default: qualified</div>

### Included schema common\_types.xsd

Namespace	No namespace
Properties	<div>attribute form default: unqualified</div> <div>element form default: qualified</div>

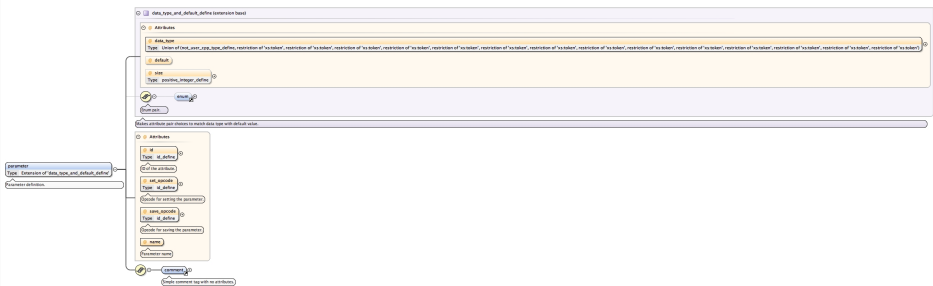
## Element(s)

### Element parameters

Namespace	No namespace
Diagram	
Properties	content: complex
Model	parameter+
Children	parameter

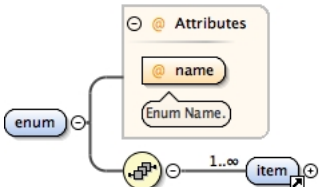
Instance	<pre>&lt;parameters opcode_base=" " parameter_base=" "   &lt;parameter data_type=" " default=" " id=" " name=" " save_opcode=" " set_opcode=" " size=" "&gt;{1,unbounded}&lt;/parameter&gt; &lt;/parameters&gt;</pre>			
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>opcode_base</b>	base_code_define	optional	
	<b>parameter_base</b>	base_code_define	optional	
Source	<pre>&lt;xs:element name="parameters"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" ref="parameter" /&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="parameter_base" type="base_code_define"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation/&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="opcode_base" type="base_code_define"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation/&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>			

## Element parameter

Namespace	No namespace			
Annotations	Parameter definition.			
Diagram				
Type	extension of data_type_and_default_define			
Type hierarchy	• data_type_and_default_define			
Properties	content:	complex		
Used by	Element	parameters		
Model	enum{0,1} , comment			
Children	comment, enum			
Instance	<pre>&lt;parameter data_type="" default="" id="" name="" save_opcode="" set_opcode="" size=""&gt;   &lt;enum name=""&gt;{0,1}&lt;/enum&gt;   &lt;comment&gt;{1,1}&lt;/comment&gt; &lt;/parameter&gt;</pre>			
Attributes	QName	Type	Use	
	data_type	union of(not_user_cpp_type_define, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token,	required	

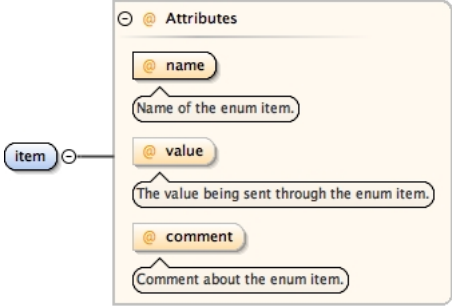
QName	Type	Use	
	restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token)		
default		optional	
id	id_define	required	
	ID of the attribute.		
name		required	
	Parameter name		
save_opcode	id_define	required	
	Opcode for saving the parameter.		
set_opcode	id_define	required	
	Opcode for setting the parameter.		
size	positive_integer_define	optional	
Source	<pre> &lt;xs:element name="parameter"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Parameter definition.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="data_type_and_default_define"&gt;         &lt;xs:sequence&gt;           &lt;xs:element ref="comment"/&gt;         &lt;/xs:sequence&gt;         &lt;xs:attribute name="id" use="required" type="id_define"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;ID of the attribute.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;         &lt;xs:attribute name="set_opcode" use="required" type="id_define"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;Opcode for setting the parameter.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;         &lt;xs:attribute name="save_opcode" use="required" type="id_define"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;Opcode for saving the parameter.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;         &lt;xs:attribute name="name" use="required"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;Parameter name&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>		

## Element enum

Namespace	No namespace	
Diagram		
Properties	content:	complex
Used by	Complex Type	data_type_and_default_define
	Element	parameter
	Element Group	type_size_choice_define
Model	item+	

Children	item			
Instance	<pre>&lt;enum name=" "&gt;   &lt;item comment=" " name=" " value=" "&gt;{1,unbounded}&lt;/item&gt; &lt;/enum&gt;</pre>			
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>name</b>		required	
	Enum Name .			
Source	<pre>&lt;xs:element name="enum"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" ref="item"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="name" use="required"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Enum Name.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>			

## Element item

Namespace	No namespace			
Diagram				
Properties	content:	complex		
Used by	Element	enum		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>comment</b>		optional	
	Comment about the enum item.			
	<b>name</b>		required	
	Name of the enum item.			
	<b>value</b>		optional	
	The value being sent through the enum item.			
Source	<pre>&lt;xs:element name="item"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="name" use="required"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Name of the enum item.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="value"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;The value being sent through the enum item.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="comment"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Comment about the enum item.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>			

## Element comment

Namespace	No namespace
-----------	--------------

Annotations	Simple comment tag with no attributes.
Diagram	
Type	xs:string
Properties	content: simple
Used by	Elements arg_define/arg, external_arg_define/arg, parameter, return
Source	<pre>&lt;xs:element name="comment" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Simple comment tag with no attributes.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element arg\_define / arg

Namespace	No namespace		
Diagram	<p>The diagram shows the structure of the <code>arg</code> element. It has four attributes: <code>name</code> (Name of the argument.), <code>pass_by</code> (Defines how the arguments are passed.), <code>comment</code> (Comments about the argument.), and <code>type_size_choice_define</code> (A choice between <code>enum</code> and <code>comment</code>). The <code>enum</code> attribute is a choice between <code>0</code> and <code>1</code>. The <code>comment</code> attribute is a simple comment tag with no attributes.</p>		
Properties	content:	complex	
Model	(enum{0,1})   comment		
Children	comment, enum		
Instance	<pre>&lt;arg comment=" " data_type=" " name=" " pass_by=" " size=" " type=" "&gt;   &lt;enum name=" "&gt;{0,1}&lt;/enum&gt;   &lt;comment&gt;{1,1}&lt;/comment&gt; &lt;/arg&gt;</pre>		
Attributes	QName	Type	Use
	comment	xs:string	optional
		Comments about the argument.	
	data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional
	name		required
		Name of the argument.	
	pass_by	pass_by_define	optional
		Defines how the arguments are passed.	
	size	xs:nonNegativeInteger	optional
		The size of the argument.	
	type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional

Source

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

## Element return

Namespace	No namespace		
Diagram			
Properties	content:	complex	
Model	(enum{0,1})   comment		
Children	comment, enum		
Instance	<pre>&lt;return comment="" data_type="" name="" pass_by="" size="" type=""&gt;   &lt;enum name=""&gt;{0,1}&lt;/enum&gt;   &lt;comment&gt;{1,1}&lt;/comment&gt; &lt;/return&gt;</pre>		
Attributes	QName	Type	Use
	comment	xs:string	optional
		Comments about the argument.	
	data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional
	name		optional
		Name of the argument.	
	pass_by	pass_by_define	optional
		Defines how the arguments are passed.	
	size	xs:nonNegativeInteger	optional

	QName	Type	Use	
		The size of the argument.		
	<b>type</b>	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
Source	<pre> &lt;xs:element name="return"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:group ref="type_size_choice_define"/&gt;       &lt;xs:element ref="comment"/&gt;     &lt;/xs:choice&gt;     &lt;xs:attribute name="name"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Name of the argument.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="pass_by" type="pass_by_define"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Defines how the arguments are passed.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="comment" type="xs:string"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Comments about the argument.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attributeGroup ref="type_size_choice_define"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>			

## Element external\_arg\_define / arg

Namespace	No namespace			
Diagram				
Properties	content:	complex		
Model	(enum{0,1})   comment			
Children	comment, enum			
Instance	<pre>&lt;arg comment=" " data_type=" " name=" " size=" " type=" "&gt;   &lt;enum name=" "&gt;{0,1}&lt;/enum&gt;   &lt;comment&gt;{1,1}&lt;/comment&gt; &lt;/arg&gt;</pre>			
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	
	<b>size</b>	xs:nonNegativeInteger	optional	
		The size of the argument.		
	<b>type</b>	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
Source	<pre> &lt;xs:element name="arg"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:group ref="type_size_choice_define"/&gt;       &lt;xs:element ref="comment"/&gt;     &lt;/xs:choice&gt;     &lt;xs:attribute name="name" use="required"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Name of the argument.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="comment" type="xs:string"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Comments about the argument.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attributeGroup ref="type_size_choice_define"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>			

## Complex Type(s)

### Complex Type data\_type\_and\_default\_define

Namespace	No namespace			
Annotations	Makes attribute pair choices to match data type with default value.			
Diagram				
Used by	Element	parameter		
Model	enum{0,1}			
Children	enum			
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>data_type</b>	union of(not_user_cpp_type_define, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token)	required	
	<b>default</b>		optional	
	<b>size</b>	positive_integer_define	optional	
Source	<pre>&lt;xs:complexType name="data_type_and_default_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Makes attribute pair choices to match data type with default value.&lt;/ xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence minOccurs="0"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Enum pair.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:element minOccurs="0" ref="enum"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="size" type="positive_integer_define" use="optional"/&gt; &lt;/xs:complexType&gt;</pre>			

```

</xs:sequence>
<xs:attribute name="data_type" use="required">
  <xs:simpleType>
    <xs:union memberTypes="not_user_cpp_type_define">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="ENUM"/>
        </xs:restriction>
      </xs:simpleType>
      <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="I8"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="U8"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="I16"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="U16"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="I32"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="U32"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="I64"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="U64"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="F32"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="F64"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="NATIVE_INT_TYPE"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="NATIVE_UINT_TYPE"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="default"/>
<xs:attribute name="size" type="positive_integer_define"/>
</xs:complexType>

```

## Simple Type(s)

### Simple Type `positive_integer_define`

Namespace	No namespace
Annotations	Positive, non-zero, whole numbers.
Diagram	
Type	restriction of <code>xs:integer</code>
Facets	<code>minInclusive</code> 1
Used by	Attribute <code>data_type_and_default_define/@size</code>
Source	<pre>&lt;xs:simpleType name="positive_integer_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Positive, non-zero, whole numbers.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:integer"&gt;     &lt;xs:minInclusive value="1"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

### Simple Type `id_define`


Namespace	No namespace
Annotations	Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".
Diagram	
Type	restriction of <code>xs:string</code>
Facets	<code>pattern</code> <code>((0?x\d+) \d+)</code>
Used by	Attributes <code>parameter/@id</code> , <code>parameter/@save_opcode</code> , <code>parameter/@set_opcode</code>
Source	<pre>&lt;xs:simpleType name="id_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Defines a ID data type. Acceptable values formats include "10" , "0xA" ,     "xA".&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:pattern value="((0?x\d+) \d+)"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

### Simple Type `base_code_define`


Namespace	No namespace
Annotations	Made for base codes, ie 0x100,0x200
Diagram	
Type	union of( <code>system_var_define</code> , restriction of <code>xs:string</code> )
Used by	Attributes <code>parameters/@opcode_base</code> , <code>parameters/@parameter_base</code>
Source	<pre>&lt;xs:simpleType name="base_code_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Made for base codes, ie 0x100,0x200&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:union memberTypes="system_var_define"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:string"&gt;         &lt;xs:pattern value="(((0?x\d+) \d+)(,?))+"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:union&gt; &lt;/xs:simpleType&gt;</pre>

```
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
```

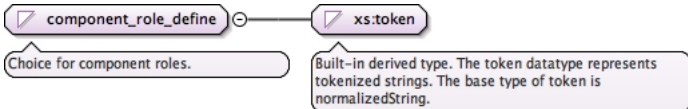
### Simple Type full\_items\_define

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

### Simple Type pass\_by\_define

Namespace	No namespace						
Annotations	Defines how the variable is being passed.						
Diagram							
Type	restriction of xs:token						
Facets	<table> <tr> <td>enumeration</td><td>reference</td></tr> <tr> <td>enumeration</td><td>value</td></tr> <tr> <td>enumeration</td><td>pointer</td></tr> </table>	enumeration	reference	enumeration	value	enumeration	pointer
enumeration	reference						
enumeration	value						
enumeration	pointer						
Used by	Attributes arg_define/arg/@pass_by, return/@pass_by						
Source	<pre>&lt;xs:simpleType name="pass_by_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Defines how the variable is being passed.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:enumeration value="reference"/&gt;     &lt;xs:enumeration value="value"/&gt;     &lt;xs:enumeration value="pointer"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>						

### Simple Type component\_role\_define

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

	enumeration	LogTextEvent
	enumeration	TimeGet
	enumeration	ParamSet
	enumeration	ParamGet
	enumeration	Telemetry
	enumeration	CmdRegistration
	enumeration	Cmd
	enumeration	CmdResponse
Source	<pre> &lt;xs:simpleType name="component_role_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Choice for component roles.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:enumeration value="LogEvent" /&gt;     &lt;xs:enumeration value="LogTextEvent" /&gt;     &lt;xs:enumeration value="TimeGet" /&gt;     &lt;xs:enumeration value="ParamSet" /&gt;     &lt;xs:enumeration value="ParamGet" /&gt;     &lt;xs:enumeration value="Telemetry" /&gt;     &lt;xs:enumeration value="CmdRegistration" /&gt;     &lt;xs:enumeration value="Cmd" /&gt;     &lt;xs:enumeration value="CmdResponse" /&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>	

### Simple Type channel\_update\_define

Namespace	No namespace				
Annotations	Choice between always and on_change. This is used in the channel 'update' tag.				
Diagram	<pre> graph LR     channel_update_define -- restriction --&gt; xs_token[xs:token]     </pre> <p>Choice between always and on_change. This is used in the channel 'update' tag.</p> <p>Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.</p>				
Type	restriction of xs:token				
Facets	<table> <tr> <td>enumeration</td> <td>always</td> </tr> <tr> <td>enumeration</td> <td>on_change</td> </tr> </table>	enumeration	always	enumeration	on_change
enumeration	always				
enumeration	on_change				
Source	<pre> &lt;xs:simpleType name="channel_update_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Choice between always and on_change. This is used in the channel 'update' tag.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:enumeration value="always" /&gt;     &lt;xs:enumeration value="on_change" /&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>				

### Simple Type severity\_define

Namespace	No namespace												
Annotations	Set of valid severity values. This is used for an event 'severity' tag.												
Diagram	<pre> graph LR     severity_define -- restriction --&gt; xs_token[xs:token]     </pre> <p>Set of valid severity values. This is used for an event 'severity' tag.</p> <p>Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.</p>												
Type	restriction of xs:token												
Facets	<table> <tr> <td>enumeration</td> <td>COMMAND</td> </tr> <tr> <td>enumeration</td> <td>ACTIVITY_LO</td> </tr> <tr> <td>enumeration</td> <td>ACTIVITY_HI</td> </tr> <tr> <td>enumeration</td> <td>WARNING_LO</td> </tr> <tr> <td>enumeration</td> <td>WARNING_HI</td> </tr> <tr> <td>enumeration</td> <td>DIAGNOSTIC</td> </tr> </table>	enumeration	COMMAND	enumeration	ACTIVITY_LO	enumeration	ACTIVITY_HI	enumeration	WARNING_LO	enumeration	WARNING_HI	enumeration	DIAGNOSTIC
enumeration	COMMAND												
enumeration	ACTIVITY_LO												
enumeration	ACTIVITY_HI												
enumeration	WARNING_LO												
enumeration	WARNING_HI												
enumeration	DIAGNOSTIC												

	enumeration FATAL
Source	<pre> &lt;xs:simpleType name="severity_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Set of valid severity values. This is used for an event 'severity' tag.&lt;/   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:enumeration value="COMMAND"/&gt;     &lt;xs:enumeration value="ACTIVITY_LO"/&gt;     &lt;xs:enumeration value="ACTIVITY_HI"/&gt;     &lt;xs:enumeration value="WARNING_LO"/&gt;     &lt;xs:enumeration value="WARNING_HI"/&gt;     &lt;xs:enumeration value="DIAGNOSTIC"/&gt;     &lt;xs:enumeration value="FATAL"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>

### Simple Type command\_kind\_define

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

### Simple Type component\_types\_define

Namespace	No namespace						
Annotations	Choice between active, passive, or queued.						
Diagram							
Type	restriction of xs:token						
Facets	<table> <tr> <td>enumeration</td> <td>active</td> </tr> <tr> <td>enumeration</td> <td>passive</td> </tr> <tr> <td>enumeration</td> <td>queued</td> </tr> </table>	enumeration	active	enumeration	passive	enumeration	queued
enumeration	active						
enumeration	passive						
enumeration	queued						
Source	<pre> &lt;xs:simpleType name="component_types_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Choice between active, passive, or queued.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:enumeration value="active"/&gt;     &lt;xs:enumeration value="passive"/&gt;     &lt;xs:enumeration value="queued"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>						

### Simple Type port\_types\_define

Namespace	No namespace
-----------	--------------

Annotations	Choice between different port types.												
Diagram	<p>The diagram shows two nodes: <b>port_types_define</b> and <b>xs:token</b>. <b>port_types_define</b> is a purple box with a choice symbol (a circle with a horizontal line) and a text box below it stating "Choice between different port types." <b>xs:token</b> is a purple box with a text box below it stating "Built-in derived type. The token datatype represents tokenized strings. The base type of token is <code>normalizedString</code>." A line connects the two nodes, indicating a restriction relationship.</p>												
Type	restriction of xs:token												
Facets	<table border="1"> <tr> <td>enumeration</td><td>input</td></tr> <tr> <td>enumeration</td><td>sync_input</td></tr> <tr> <td>enumeration</td><td>guarded_input</td></tr> <tr> <td>enumeration</td><td>async_input</td></tr> <tr> <td>enumeration</td><td>model_input</td></tr> <tr> <td>enumeration</td><td>output</td></tr> </table>	enumeration	input	enumeration	sync_input	enumeration	guarded_input	enumeration	async_input	enumeration	model_input	enumeration	output
enumeration	input												
enumeration	sync_input												
enumeration	guarded_input												
enumeration	async_input												
enumeration	model_input												
enumeration	output												
Source	<pre> &lt;xs:simpleType name="port_types_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Choice between different port types.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:enumeration value="input"/&gt;     &lt;xs:enumeration value="sync_input"/&gt;     &lt;xs:enumeration value="guarded_input"/&gt;     &lt;xs:enumeration value="async_input"/&gt;     &lt;xs:enumeration value="model_input"/&gt;     &lt;xs:enumeration value="output"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>												

## 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	<pre> classDiagram     class id_or_system_var_define {         "Data types for items that can either be numbers or references to system variables that have numbers."     }     class system_var_define {         "Data type for items that are system variables."     }     class id_define {         "Defines a ID data type. Acceptable values formats include \"10\", \"0xA\", \"xA\"."     }     id_or_system_var_define -- &gt; system_var_define     id_or_system_var_define -- &gt; id_define     </pre>
Type	union of(system_var_define, id_define)
Source	<pre> &lt;xs:simpleType name="id_or_system_var_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Data types for items that can either be numbers or references to system     variables that have numbers.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:union memberTypes="system_var_define id_define"/&gt; &lt;/xs:simpleType&gt; </pre>

## Simple Type system\_var\_define

Namespace	No namespace
Annotations	Data type for items that are system variables.
Diagram	
Type	restriction of xs:string
Facets	pattern <code>\$(\w _ \\- )+</code>
Source	<pre> &lt;xs:simpleType name="system_var_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Data type for items that are system variables.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:pattern value="\$(\w _ \\- )+"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>

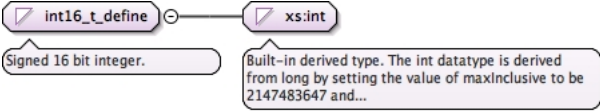
## Simple Type int8\_t\_define

Namespace	No namespace		
Annotations	Signed 8 bit integer.		
Diagram			
Type	restriction of xs:int		
Facets	maxInclusive	127	
	minInclusive	-128	
Used by	Simple Type	I8_define	
Source	<pre>&lt;xs:simpleType name="int8_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Signed 8 bit integer.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:int"&gt;     &lt;xs:minInclusive value="-128"/&gt;     &lt;xs:maxInclusive value="127"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>		

## Simple Type uint8\_t\_define

Namespace	No namespace		
Annotations	Unsigned 8 bit integer		
Diagram			
Type	restriction of xs:unsignedByte		
Facets	maxInclusive	255	
	minInclusive	0	
Used by	Simple Type	U8_define	
Source	<pre>&lt;xs:simpleType name="uint8_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Unsigned 8 bit integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:unsignedByte"&gt;     &lt;xs:minInclusive value="0"/&gt;     &lt;xs:maxInclusive value="255"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>		

## Simple Type int16\_t\_define

Namespace	No namespace		
Annotations	Signed 16 bit integer.		
Diagram			
Type	restriction of xs:int		
Facets	maxInclusive	32767	
	minInclusive	-32768	
Used by	Simple Type	I16_define	
Source	<pre>&lt;xs:simpleType name="int16_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Signed 16 bit integer.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:simpleType&gt;</pre>		



```
<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			
Type	restriction of xs:int		
Facets	maxInclusive	65535	
	minInclusive	0	
Used by	Simple Type	U16_define	
Source	<pre>&lt;xs:simpleType name="uint16_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Unsigned 16 bit integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:int"&gt;     &lt;xs:minInclusive value="0"/&gt;     &lt;xs:maxInclusive value="65535"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>		

## Simple Type int32\_t\_define

Namespace	No namespace		
Annotations	Signed 32 bit integer.		
Diagram	<p>The diagram illustrates the relationship between the <code>int32_t_define</code> simple type and the <code>xs:integer</code> built-in type. <code>int32_t_define</code> is shown as a restriction of <code>xs:integer</code>, specifically for the range from -2147483648 to 2147483647. A note explains that <code>xs:integer</code> is a built-in derived type from decimal, derived by fixing the value of <code>fractionDigits</code> to be 0.</p>		
Type	restriction of xs:integer		
Facets	maxInclusive	2147483647	
	minInclusive	-2147483648	
Used by	Simple Types	I32_define, NATIVE_INT_TYPE_define	
Source	<pre>&lt;xs:simpleType name="int32_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Signed 32 bit integer.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:integer"&gt;     &lt;xs:minInclusive value="-2147483648"/&gt;     &lt;xs:maxInclusive value="2147483647"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>		

## Simple Type uint32\_t\_define

Namespace	No namespace		
Annotations	Unsigned 32 bit integer		
Diagram			
Type	restriction of xs:integer		
Facets	maxInclusive	4294967295	
	minInclusive	0	

Used by	Simple Types NATIVE_UINT_TYPE_define, U32_define
Source	<pre> &lt;xs:simpleType name="uint32_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Unsigned 32 bit integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:integer"&gt;     &lt;xs:minInclusive value="0"/&gt;     &lt;xs:maxInclusive value="4294967295"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>

### Simple Type int64\_t\_define

Namespace	No namespace				
Annotations	Signed 64 bit integer.				
Diagram					
Type	restriction of xs:integer				
Facets	<table> <tr> <td>maxInclusive</td><td>9223372036854775807</td></tr> <tr> <td>minInclusive</td><td>-9223372036854775808</td></tr> </table>	maxInclusive	9223372036854775807	minInclusive	-9223372036854775808
maxInclusive	9223372036854775807				
minInclusive	-9223372036854775808				
Used by	Simple Type I64_define				
Source	<pre> &lt;xs:simpleType name="int64_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Signed 64 bit integer.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:integer"&gt;     &lt;xs:minInclusive value="-9223372036854775808"/&gt;     &lt;xs:maxInclusive value="9223372036854775807"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>				

### Simple Type uint64\_t\_define

Namespace	No namespace				
Annotations	Unsigned 64 bit integer				
Diagram					
Type	restriction of xs:integer				
Facets	<table> <tr> <td>maxInclusive</td><td>18446744073709551615</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	18446744073709551615	minInclusive	0
maxInclusive	18446744073709551615				
minInclusive	0				
Used by	Simple Type U64_define				
Source	<pre> &lt;xs:simpleType name="uint64_t_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Unsigned 64 bit integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:integer"&gt;     &lt;xs:minInclusive value="0"/&gt;     &lt;xs:maxInclusive value="18446744073709551615"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>				

### 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	

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

### Simple Type NATIVE\_INT\_TYPE\_define

Namespace	No namespace				
Annotations	native integer type declaration				
Diagram					
Type	int32_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:integer           <ul style="list-style-type: none"> <li>int32_t_define               <ul style="list-style-type: none"> <li>NATIVE_INT_TYPE_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>2147483647</td></tr> <tr> <td>minInclusive</td><td>-2147483648</td></tr> </table>	maxInclusive	2147483647	minInclusive	-2147483648
maxInclusive	2147483647				
minInclusive	-2147483648				
Source	<pre>&lt;xs:simpleType name="NATIVE_INT_TYPE_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;native integer type declaration&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="int32_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

### Simple Type NATIVE\_UINT\_TYPE\_define

Namespace	No namespace				
Annotations	native unsigned integer type declaration				
Diagram					
Type	uint32_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:integer           <ul style="list-style-type: none"> <li>uint32_t_define               <ul style="list-style-type: none"> <li>NATIVE_UINT_TYPE_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>4294967295</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	4294967295	minInclusive	0
maxInclusive	4294967295				
minInclusive	0				
Source	<pre>&lt;xs:simpleType name="NATIVE_UINT_TYPE_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;native unsigned integer type declaration&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="uint32_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

### Simple Type I8\_define

Namespace	No namespace
Annotations	8-bit signed integer
Diagram	
Type	int8_t_define
Type hierarchy	<ul style="list-style-type: none"> <li>xs:int           <ul style="list-style-type: none"> <li>int8_t_define</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• int8_t_define</li> <li>• I8_define</li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>127</td></tr> <tr> <td>minInclusive</td><td>-128</td></tr> </table>	maxInclusive	127	minInclusive	-128
maxInclusive	127				
minInclusive	-128				
Source	<pre>&lt;xs:simpleType name="I8_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;8-bit signed integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="int8_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

## Simple Type U8\_define

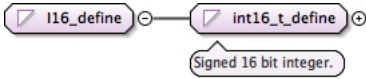
Namespace	No namespace				
Annotations	8-bit unsigned integer				
Diagram	<pre> graph LR     U8_define[U8_define] --- restriction  uint8_t_define[uint8_t_define]     U8_define --- U8_label[8-bit unsigned integer]     uint8_t_define --- U8_label2[Unsigned 8 bit integer]   </pre>				
Type	uint8_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:unsignedByte</li> <li>• uint8_t_define</li> <li>• U8_define</li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>255</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	255	minInclusive	0
maxInclusive	255				
minInclusive	0				
Used by	Simple Type BYTE_define				
Source	<pre>&lt;xs:simpleType name="U8_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;8-bit unsigned integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="uint8_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

## Simple Type BYTE\_define

Namespace	No namespace				
Annotations	byte type				
Diagram	<pre> graph LR     BYTE_define[BYTE_define] --- restriction  U8_define[U8_define]     BYTE_define --- B_label[byte type]     U8_define --- U8_label[8-bit unsigned integer]   </pre>				
Type	U8_define				
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:unsignedByte</li> <li>• uint8_t_define</li> <li>• U8_define</li> <li>• BYTE_define</li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>255</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	255	minInclusive	0
maxInclusive	255				
minInclusive	0				
Source	<pre>&lt;xs:simpleType name="BYTE_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;byte type&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="U8_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

## Simple Type I16\_define

Namespace	No namespace
-----------	--------------

Diagram					
Type	int16_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:int <ul style="list-style-type: none"> <li>int16_t_define <ul style="list-style-type: none"> <li>I16_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>32767</td></tr> <tr> <td>minInclusive</td><td>-32768</td></tr> </table>	maxInclusive	32767	minInclusive	-32768
maxInclusive	32767				
minInclusive	-32768				
Source	<pre>&lt;xs:simpleType name="I16_define"&gt;   &lt;xs:restriction base="int16_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

### Simple Type U16\_define

Namespace	No namespace				
Annotations	16-bit unsigned integer				
Diagram					
Type	uint16_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:int <ul style="list-style-type: none"> <li>uint16_t_define <ul style="list-style-type: none"> <li>U16_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>65535</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	65535	minInclusive	0
maxInclusive	65535				
minInclusive	0				
Source	<pre>&lt;xs:simpleType name="U16_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;16-bit unsigned integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="uint16_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

### Simple Type I32\_define

Namespace	No namespace				
Annotations	32-bit signed integer				
Diagram					
Type	int32_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:integer <ul style="list-style-type: none"> <li>int32_t_define <ul style="list-style-type: none"> <li>I32_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>2147483647</td></tr> <tr> <td>minInclusive</td><td>-2147483648</td></tr> </table>	maxInclusive	2147483647	minInclusive	-2147483648
maxInclusive	2147483647				
minInclusive	-2147483648				
Source	<pre>&lt;xs:simpleType name="I32_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;32-bit signed integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="int32_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

### Simple Type U32\_define

Namespace	No namespace
-----------	--------------

Annotations	16-bit unsigned integer				
Diagram					
Type	uint32_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:integer <ul style="list-style-type: none"> <li>uint32_t_define <ul style="list-style-type: none"> <li>U32_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>4294967295</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	4294967295	minInclusive	0
maxInclusive	4294967295				
minInclusive	0				
Source	<pre>&lt;xs:simpleType name="U32_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;16-bit unsigned integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="uint32_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

## Simple Type I64\_define

Namespace	No namespace				
Annotations	64-bit unsigned integer				
Diagram					
Type	int64_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:integer <ul style="list-style-type: none"> <li>int64_t_define <ul style="list-style-type: none"> <li>I64_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>9223372036854775807</td></tr> <tr> <td>minInclusive</td><td>-9223372036854775808</td></tr> </table>	maxInclusive	9223372036854775807	minInclusive	-9223372036854775808
maxInclusive	9223372036854775807				
minInclusive	-9223372036854775808				
Source	<pre>&lt;xs:simpleType name="I64_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;64-bit unsigned integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="int64_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

## Simple Type U64\_define

Namespace	No namespace				
Annotations	64-bit unsigned integer				
Diagram					
Type	uint64_t_define				
Type hierarchy	<ul style="list-style-type: none"> <li>xs:integer <ul style="list-style-type: none"> <li>uint64_t_define <ul style="list-style-type: none"> <li>U64_define</li> </ul> </li> </ul> </li> </ul>				
Facets	<table> <tr> <td>maxInclusive</td><td>18446744073709551615</td></tr> <tr> <td>minInclusive</td><td>0</td></tr> </table>	maxInclusive	18446744073709551615	minInclusive	0
maxInclusive	18446744073709551615				
minInclusive	0				
Source	<pre>&lt;xs:simpleType name="U64_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;64-bit unsigned integer&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="uint64_t_define"/&gt; &lt;/xs:simpleType&gt;</pre>				

## Simple Type F32\_define

Namespace	No namespace
Annotations	32 bit float
Diagram	
Type	xs:float
Source	<pre>&lt;xs:simpleType name="F32_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;32 bit float&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:float"/&gt; &lt;/xs:simpleType&gt;</pre>

## Simple Type F64\_define

Namespace	No namespace
Annotations	64 bit float
Diagram	
Type	xs:double
Source	<pre>&lt;xs:simpleType name="F64_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;64 bit float&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:double"/&gt; &lt;/xs:simpleType&gt;</pre>

## Attribute(s)

### Attribute item / @name

Namespace	No namespace
Annotations	Name of the enum item.
Properties	use: required
Used by	Element item
Source	<pre>&lt;xs:attribute name="name" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Name of the enum item.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute item / @value

Namespace	No namespace
Annotations	The value being sent through the enum item.
Used by	Element item
Source	<pre>&lt;xs:attribute name="value"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The value being sent through the enum item.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute item / @comment

Namespace	No namespace
Annotations	Comment about the enum item.
Used by	Element item

Source	<pre>&lt;xs:attribute name="comment"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Comment about the enum item.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>
--------	---

### Attribute enum / @name

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

### Attribute data\_type\_and\_default\_define / @data\_type

Namespace	No namespace
Type	union of(not_user_cpp_type_define, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token)
Properties	use: required
Used by	Complex Type data_type_and_default_define
Source	<pre>&lt;xs:attribute name="data_type" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:union memberTypes="not_user_cpp_type_define"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="ENUM"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="string"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="I8"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="U8"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="I16"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="U16"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="I32"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="U32"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="I64"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:union&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>



```

</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:token">
    <xs:enumeration value="U64" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:token">
    <xs:enumeration value="F32" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:token">
    <xs:enumeration value="F64" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:token">
    <xs:enumeration value="NATIVE_INT_TYPE" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:token">
    <xs:enumeration value="NATIVE_UINT_TYPE" />
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
</xs:attribute>

```

#### Attribute data\_type\_and\_default\_define / @default

Namespace	No namespace
Used by	Complex Type data_type_and_default_define
Source	<code>&lt;xs:attribute name="default" /&gt;</code>

#### Attribute data\_type\_and\_default\_define / @size

Namespace	No namespace
Type	positive_integer_define
Properties	content: simple
Facets	minInclusive 1
Used by	Complex Type data_type_and_default_define
Source	<code>&lt;xs:attribute name="size" type="positive_integer_define" /&gt;</code>

#### Attribute parameter / @id

Namespace	No namespace
Annotations	ID of the attribute.
Type	id_define
Properties	use: required
Facets	pattern ((0?x\d+) \d+)
Used by	Element parameter
Source	<pre> &lt;xs:attribute name="id" use="required" type="id_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;ID of the attribute.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

#### Attribute parameter / @set\_opcode

Namespace	No namespace
Annotations	Opcode for setting the parameter.
Type	id_define

Properties	use: required
Facets	pattern ((0?x\d+) \d+)
Used by	Element parameter
Source	<pre>&lt;xs:attribute name="set_opcode" use="required" type="id_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Opcode for setting the parameter.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute parameter / @save\_opcode

Namespace	No namespace
Annotations	Opcode for saving the parameter.
Type	id_define
Properties	use: required
Facets	pattern ((0?x\d+) \d+)
Used by	Element parameter
Source	<pre>&lt;xs:attribute name="save_opcode" use="required" type="id_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Opcode for saving the parameter.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute parameter / @name

Namespace	No namespace
Annotations	Parameter name
Properties	use: required
Used by	Element parameter
Source	<pre>&lt;xs:attribute name="name" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Parameter name&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute parameters / @parameter\_base

Namespace	No namespace
Annotations	
Type	base_code_define
Properties	content: simple
Used by	Element parameters
Source	<pre>&lt;xs:attribute name="parameter_base" type="base_code_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation/&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute parameters / @opcode\_base

Namespace	No namespace
Annotations	
Type	base_code_define
Properties	content: simple
Used by	Element parameters
Source	<pre>&lt;xs:attribute name="opcode_base" type="base_code_define"&gt;   &lt;xs:annotation&gt;</pre>

```
<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 <code>arg_define/arg</code>
Source	<pre>&lt;xs:attribute name="name" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Name of the argument.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute `arg_define` / `arg` / `@pass_by`

Namespace	No namespace
Annotations	Defines how the arguments are passed.
Type	<code>pass_by_define</code>
Properties	content: simple
Facets	enumeration reference
	enumeration value
	enumeration pointer
Used by	Element <code>arg_define/arg</code>
Source	<pre>&lt;xs:attribute name="pass_by" type="pass_by_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Defines how the arguments are passed.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute `arg_define` / `arg` / `@comment`

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

### Attribute `type_size_choice_define` / `@data_type`

Namespace	No namespace
Type	union of( <code>xs:string</code> , restriction of <code>xs:token</code> , restriction of <code>xs:token</code> )
Properties	content: simple
Used by	Attribute Group <code>type_size_choice_define</code>
Source	<pre>&lt;xs:attribute name="data_type"&gt;   &lt;xs:simpleType&gt;     &lt;xs:union memberTypes="xs:string"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="string"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;</pre>

```

        <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	<pre> &lt;xs:attribute name="type"&gt;   &lt;xs:simpleType&gt;     &lt;xs:union memberTypes="xs:string"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="string"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:token"&gt;           &lt;xs:enumeration value="ENUM"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:union&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>

### 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	<pre> &lt;xs:attribute name="size" type="xs:nonNegativeInteger"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The size of the argument.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

### Attribute return / @name

Namespace	No namespace
Annotations	Name of the argument.
Used by	Element return
Source	<pre> &lt;xs:attribute name="name"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Name of the argument.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

### 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	<pre>&lt;xs:attribute name="pass_by" type="pass_by_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Defines how the arguments are passed.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute return / @comment

Namespace	No namespace
Annotations	Comments about the argument.
Type	xs:string
Properties	content: simple
Used by	Element return
Source	<pre>&lt;xs:attribute name="comment" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Comments about the argument.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### Attribute external\_arg\_define / arg / @name

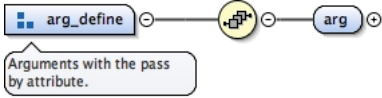
Namespace	No namespace
Annotations	Name of the argument.
Properties	use: required
Used by	Element external_arg_define/arg
Source	<pre>&lt;xs:attribute name="name" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Name of the argument.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;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

### 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	<pre>&lt;xs:attribute name="comment" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Comments about the argument.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

## Element Group(s)

### Element Group arg\_define

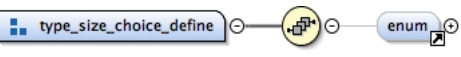
Namespace	No namespace
Annotations	Arguments with the pass by attribute.
Diagram	
Model	arg
Children	arg
Source	<pre>&lt;xs:group name="arg_define"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Arguments with the pass by attribute.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;</pre>

```

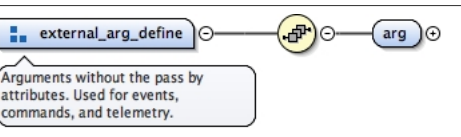
<xs:sequence>
  <xs:element name="arg">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="type_size_choice_define"/>
        <xs:element ref="comment"/>
      </xs:choice>
      <xs:attribute name="name" use="required">
        <xs:annotation>
          <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	<pre> &lt;xs:group name="type_size_choice_define"&gt;   &lt;xs:sequence&gt;     &lt;xs:element minOccurs="0" ref="enum"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:group&gt; </pre>

## Element Group external\_arg\_define

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

```

        </xs:annotation>
      </xs:attribute>
      <xs:attributeGroup ref="type_size_choice_define" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:group>

```

## Attribute Group(s)

### Attribute Group type\_size\_choice\_define

Namespace	No namespace			
Diagram				
Used by	Elements arg_define/arg, external_arg_define/arg, 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	<pre>&lt;xs:attributeGroup name="type_size_choice_define"&gt;   &lt;xs:attribute name="data_type"&gt;     &lt;xs:simpleType&gt;       &lt;xs:union memberTypes="xs:string"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:token"&gt;             &lt;xs:enumeration value="string"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:token"&gt;             &lt;xs:enumeration value="ENUM"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:union&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="type"&gt;     &lt;xs:simpleType&gt;       &lt;xs:union memberTypes="xs:string"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:token"&gt;             &lt;xs:enumeration value="string"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:token"&gt;             &lt;xs:enumeration value="ENUM"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:union&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="size" type="xs:nonNegativeInteger"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The size of the argument.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:attributeGroup&gt;</pre>			