

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	<table> <tr> <td>attribute form default:</td><td>unqualified</td></tr> <tr> <td>element form default:</td><td>qualified</td></tr> </table>	attribute form default:	unqualified	element form default:	qualified
attribute form default:	unqualified				
element form default:	qualified				

Included schema common_elements.xsd

Namespace	No namespace				
Properties	<table> <tr> <td>attribute form default:</td><td>unqualified</td></tr> <tr> <td>element form default:</td><td>qualified</td></tr> </table>	attribute form default:	unqualified	element form default:	qualified
attribute form default:	unqualified				
element form default:	qualified				

Included schema common_types.xsd

Namespace	No namespace				
Properties	<table> <tr> <td>attribute form default:</td><td>unqualified</td></tr> <tr> <td>element form default:</td><td>qualified</td></tr> </table>	attribute form default:	unqualified	element form default:	qualified
attribute form default:	unqualified				
element form default:	qualified				


Element(s)

Element parameters

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

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

Element parameter

Namespace	No namespace			
Annotations	Parameter definition.			
Diagram				
Type	extension of data_type_and_default_define			
Type hierarchy	<ul style="list-style-type: none">data_type_and_default_define			
Properties	content:	complex		
Used by	Element	parameters		
Model	enum{0,1} , comment			
Children	comment, enum			
Instance	<pre><parameter data_type=" " default=" " id=" " name=" " save_opcode=" " set_opcode=" " size=" "> <enum name=" ">{0,1}</enum> <comment>{1,1}</comment> </parameter></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	

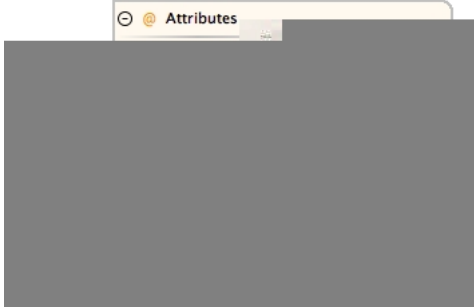
QName	Type	Use	
	restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token)		
default		optional	
id	id_define	required	
	ID of the attribute.		
name		required	
	Parameter name		
save_opcode	id_define	required	
	Opcode for saving the parameter.		
set_opcode	id_define	required	
	Opcode for setting the parameter.		
size	positive_integer_define	optional	
Source	<pre> <xs:element name="parameter"> <xs:annotation> <xs:documentation>Parameter definition.</xs:documentation> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="data_type_and_default_define"> <xs:sequence> <xs:element ref="comment"/> </xs:sequence> <xs:attribute name="id" use="required" type="id_define"> <xs:annotation> <xs:documentation>ID of the attribute.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="set_opcode" use="required" type="id_define"> <xs:annotation> <xs:documentation>Opcode for setting the parameter.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="save_opcode" use="required" type="id_define"> <xs:annotation> <xs:documentation>Opcode for saving the parameter.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Parameter name</xs:documentation> </xs:annotation> </xs:attribute> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element> </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><enum name=" "> <item comment=" " name=" " value=" ">{1,unbounded}</item> </enum></pre>			
Attributes	QName	Type	Use	
	name		required	
	Enum Name .			
Source	<pre><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></pre>			

Element item

Namespace	No namespace			
Diagram				
Properties	content:	complex		
Used by	Element	enum		
Attributes	QName	Type	Use	
	comment		optional	
	Comment about the enum item.			
	name		required	
	Name of the enum item.			
	value		optional	
	The value being sent through the enum item.			
Source	<pre><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></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><xs:element name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Simple comment tag with no attributes.</xs:documentation> </xs:annotation> </xs:element></pre>

Element arg_define / arg

Namespace	No namespace																																														
Diagram																																															
Properties	content:	complex																																													
Model	(enum{0,1}) comment																																														
Children	comment, enum																																														
Instance	<pre><arg comment="" data_type="" name="" pass_by="" size="" type=""> <enum name="">{0,1}</enum> <comment>{1,1}</comment> </arg></pre>																																														
Attributes	<table><tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr><tr><td>comment</td><td>xs:string</td><td>optional</td><td></td></tr><tr><td></td><td colspan="3">Comments about the argument.</td></tr><tr><td>data_type</td><td>union of(xs:string, restriction of xs:token, restriction of xs:token)</td><td>optional</td><td></td></tr><tr><td>name</td><td></td><td>required</td><td></td></tr><tr><td></td><td colspan="3">Name of the argument.</td></tr><tr><td>pass_by</td><td>pass_by_define</td><td>optional</td><td></td></tr><tr><td></td><td colspan="3">Defines how the arguments are passed.</td></tr><tr><td>size</td><td>xs:nonNegativeInteger</td><td>optional</td><td></td></tr><tr><td></td><td colspan="3">The size of the argument.</td></tr><tr><td>type</td><td>union of(xs:string, restriction of xs:token, restriction of xs:token)</td><td>optional</td><td></td></tr></table>	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			
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

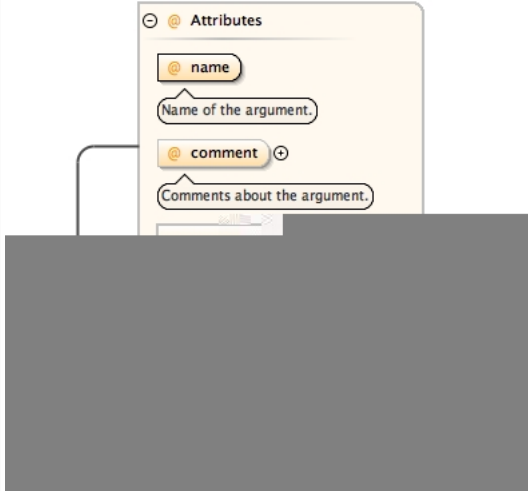
```
<xs:element name="arg">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="type_size_choice_define"/>
      <xs:element ref="comment"/>
    </xs:choice>
    <xs:attribute name="name" use="required">
      <xs:annotation>
        <xs:documentation>Name of the 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><return comment="" data_type="" name="" pass_by="" size="" type=""> <enum name="">{0,1}</enum> <comment>{1,1}</comment> </return></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.		
	type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
Source	<pre> <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> </pre>			


Element external_arg_define / arg

Namespace	No namespace			
Diagram				
Properties	content: complex			
Model	(enum{0,1}) comment			
Children	comment, enum			
Instance	<pre><arg comment="" data_type="" name="" size="" type=""> <enum name="">{0,1}</enum> <comment>{1,1}</comment> </arg></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	
	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> <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> </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	<table><thead><tr><th>QName</th><th>Type</th><th>Use</th></tr></thead><tbody><tr><td>data_type</td><td>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)</td><td>required</td></tr><tr><td>default</td><td></td><td>optional</td></tr><tr><td>size</td><td>positive_integer_define</td><td>optional</td></tr></tbody></table>	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, restriction of xs:token, restriction of xs:token, restriction of xs:token)	required	default		optional	size	positive_integer_define	optional			
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, restriction of xs:token, restriction of xs:token, restriction of xs:token)	required														
default		optional														
size	positive_integer_define	optional														
Source	<pre><xs:complexType name="data_type_and_default_define"> <xs:annotation> <xs:documentation>Makes attribute pair choices to match data type with default value.</ xs:documentation> </xs:annotation> <xs:sequence minOccurs="0"> <xs:annotation> <xs:documentation>Enum pair.</xs:documentation> </xs:annotation> <xs:element minOccurs="0" ref="enum"/> </xs:sequence> </xs:complexType></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><xs:simpleType name="positive_integer_define"> <xs:annotation> <xs:documentation>Positive, non-zero, whole numbers.</xs:documentation> </xs:annotation> <xs:restriction base="xs:integer"> <xs:minInclusive value="1"/> </xs:restriction> </xs:simpleType></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><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></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><xs:simpleType name="base_code_define"> <xs:annotation> <xs:documentation>Made for base codes, ie 0x100,0x200</xs:documentation> </xs:annotation> <xs:union memberTypes="system_var_define"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:pattern value="(((0?x\d+) \d+)(,?))+"/> </xs:restriction> </xs:simpleType> </xs:union> </xs:simpleType></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 border="1"> <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> <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>						

Simple Type pass_by_define

Namespace	No namespace						
Annotations	Defines how the variable is being passed.						
Diagram							
Type	restriction of xs:token						
Facets	<table border="1"> <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> <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>						

Simple Type component_role_define

Namespace	No namespace		
Annotations	Choice for component roles.		
Diagram			
Type	restriction of xs:token		
Facets	<table border="1"> <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> <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> </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					
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> <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> </pre>				

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	<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><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></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><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></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><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></pre>						

Simple Type port_types_define

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

Annotations	Choice between different port types.												
Diagram	<p>The diagram shows a box labeled 'port_types_define' with a choice symbol (a circle with a vertical line). It is connected to a box labeled 'xs.token' with a 'Built-in derived type. The token datatype represents' label. A callout points to the choice symbol with the text 'Choice between different port types'.</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> <xs:simpleType name="port_types_define"> <xs:annotation> <xs:documentation>Choice between different port types.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="input"/> <xs:enumeration value="sync_input"/> <xs:enumeration value="guarded_input"/> <xs:enumeration value="async_input"/> <xs:enumeration value="model_input"/> <xs:enumeration value="output"/> </xs:restriction> </xs:simpleType> </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	<p>The diagram shows a box labeled 'id_or_system_var_define' with a union symbol (a circle with a horizontal line). It is connected to two boxes: 'system_var_define' (with a 'Data type for items that are system variables.' label) and 'id_define' (with a 'Define' label). A callout points to the union symbol with the text 'Data types for items that can either be numbers or references to system variables that have numbers.'</p>
Type	union of(system_var_define, id_define)
Source	<pre> <xs:simpleType name="id_or_system_var_define"> <xs:annotation> <xs:documentation>Data types for items that can either be numbers or references to system variables that have numbers.</xs:documentation> </xs:annotation> <xs:union memberTypes="system_var_define id_define"/> </xs:simpleType> </pre>

Simple Type system_var_define

Namespace	No namespace		
Annotations	Data type for items that are system variables.		
Diagram	<p>The diagram shows a box labeled 'system_var_define' with a restriction symbol (a circle with a diagonal line). It is connected to a box labeled 'xs:string' with a 'Built-in primitive type. The string datatype represents character strings in XML.' label. A callout points to the restriction symbol with the text 'Data type for items that are system variables.'</p>		
Type	restriction of xs:string		
Facets	<table border="1"> <tr><td>pattern</td><td>\$[\w _ -]+</td></tr> </table>	pattern	\$[\w _ -]+
pattern	\$[\w _ -]+		
Source	<pre> <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> </pre>		

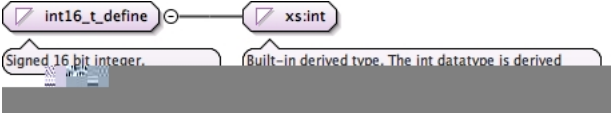
Simple Type int8_t_define

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

Simple Type uint8_t_define

Namespace	No namespace		
Annotations	Unsigned 8 bit integer		
Diagram	<pre>graph LR; A[uint8_t_define] --- B[xs:unsignedByte];</pre>		
Type	restriction of xs:unsignedByte		
Facets	maxInclusive	255	
	minInclusive	0	
Used by	Simple Type	U8_define	
Source	<pre><xs:simpleType name="uint8_t_define"> <xs:annotation> <xs:documentation>Unsigned 8 bit integer</xs:documentation> </xs:annotation> <xs:restriction base="xs:unsignedByte"> <xs:minInclusive value="0"/> <xs:maxInclusive value="255"/> </xs:restriction> </xs:simpleType></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><xs:simpleType name="int16_t_define"> <xs:annotation> <xs:documentation>Signed 16 bit integer.</xs:documentation> </xs:annotation> </xs:simpleType></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	<p>The diagram illustrates the relationship between the <code>uint16_t_define</code> simple type and the <code>xs:int</code> built-in type. <code>uint16_t_define</code> is shown as a restriction of <code>xs:int</code>. A callout box explains that <code>xs:int</code> is a built-in derived type derived from <code>long</code> by setting <code>maxInclusive</code> to <code>2147483647</code>.</p>		
Type	restriction of xs:int		
Facets	maxInclusive	65535	
	minInclusive	0	
Used by	Simple Type	U16_define	
Source	<pre><xs:simpleType name="uint16_t_define"> <xs:annotation> <xs:documentation>Unsigned 16 bit integer</xs:documentation> </xs:annotation> <xs:restriction base="xs:int"> <xs:minInclusive value="0"/> <xs:maxInclusive value="65535"/> </xs:restriction> </xs:simpleType></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>, indicated by a line with an open circle at the <code>int32_t_define</code> end. Below <code>int32_t_define</code> is a callout box stating "Signed 32 bit integer." Below <code>xs:integer</code> is a callout box stating "Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This..."</p>		
Type	restriction of xs:integer		
Facets	maxInclusive	2147483647	
	minInclusive	-2147483648	
Used by	Simple Types	I32_define, NATIVE_INT_TYPE_define	
Source	<pre><xs:simpleType name="int32_t_define"> <xs:annotation> <xs:documentation>Signed 32 bit integer.</xs:documentation> </xs:annotation> <xs:restriction base="xs:integer"> <xs:minInclusive value="-2147483648"/> <xs:maxInclusive value="2147483647"/> </xs:restriction> </xs:simpleType></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> <xs:simpleType name="uint32_t_define"> <xs:annotation> <xs:documentation>Unsigned 32 bit integer</xs:documentation> </xs:annotation> <xs:restriction base="xs:integer"> <xs:minInclusive value="0"/> <xs:maxInclusive value="4294967295"/> </xs:restriction> </xs:simpleType> </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> <xs:simpleType name="int64_t_define"> <xs:annotation> <xs:documentation>Signed 64 bit integer.</xs:documentation> </xs:annotation> <xs:restriction base="xs:integer"> <xs:minInclusive value="-9223372036854775808"/> <xs:maxInclusive value="9223372036854775807"/> </xs:restriction> </xs:simpleType> </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> <xs:simpleType name="uint64_t_define"> <xs:annotation> <xs:documentation>Unsigned 64 bit integer</xs:documentation> </xs:annotation> <xs:restriction base="xs:integer"> <xs:minInclusive value="0"/> <xs:maxInclusive value="18446744073709551615"/> </xs:restriction> </xs:simpleType> </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> <xs:simpleType name="not_user_cpp_type_define"> <xs:annotation> <xs:documentation>Ensures data is not of the names of any other user defined C++ name.</ </xs:annotation> <xs:restriction base="xs:string"/> </xs:simpleType> </pre>

Simple Type NATIVE_INT_TYPE_define

Namespace	No namespace				
Annotations	native integer type declaration				
Diagram	<pre> graph LR NATIVE_INT_TYPE_define --> int32_t_define style NATIVE_INT_TYPE_define fill:#f9f9f9,stroke:#333,stroke-width:1px style int32_t_define fill:#f9f9f9,stroke:#333,stroke-width:1px </pre>				
Type	int32_t_define				
Type hierarchy	<ul style="list-style-type: none"> xs:integer <ul style="list-style-type: none"> int32_t_define <ul style="list-style-type: none"> NATIVE_INT_TYPE_define 				
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> <xs:simpleType name="NATIVE_INT_TYPE_define"> <xs:annotation> <xs:documentation>native integer type declaration</xs:documentation> </xs:annotation> <xs:restriction base="int32_t_define"/> </xs:simpleType> </pre>				

Simple Type NATIVE_UINT_TYPE_define

Namespace	No namespace				
Annotations	native unsigned integer type declaration				
Diagram	<pre> graph LR NATIVE_UINT_TYPE_define --> uint32_t_define style NATIVE_UINT_TYPE_define fill:#f9f9f9,stroke:#333,stroke-width:1px style uint32_t_define fill:#f9f9f9,stroke:#333,stroke-width:1px </pre>				
Type	uint32_t_define				
Type hierarchy	<ul style="list-style-type: none"> xs:integer <ul style="list-style-type: none"> uint32_t_define <ul style="list-style-type: none"> NATIVE_UINT_TYPE_define 				
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> <xs:simpleType name="NATIVE_UINT_TYPE_define"> <xs:annotation> <xs:documentation>native unsigned integer type declaration</xs:documentation> </xs:annotation> <xs:restriction base="uint32_t_define"/> </xs:simpleType> </pre>				

Simple Type I8_define

Namespace	No namespace
Annotations	8-bit signed integer
Diagram	<pre> graph LR I8_define --> int8_t_define style I8_define fill:#f9f9f9,stroke:#333,stroke-width:1px style int8_t_define fill:#f9f9f9,stroke:#333,stroke-width:1px </pre>
Type	int8_t_define
Type hierarchy	<ul style="list-style-type: none"> xs:int <ul style="list-style-type: none"> int8_t_define

	<ul style="list-style-type: none"> • int8_t_define • I8_define 				
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><xs:simpleType name="I8_define"> <xs:annotation> <xs:documentation>8-bit signed integer</xs:documentation> </xs:annotation> <xs:restriction base="int8_t_define"/> </xs:simpleType></pre>				

Simple Type U8_define

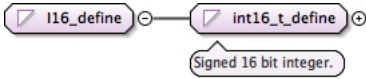
Namespace	No namespace				
Annotations	8-bit unsigned integer				
Diagram	<pre> graph LR U8_define -- restriction --> uint8_t_define </pre>				
Type	uint8_t_define				
Type hierarchy	<ul style="list-style-type: none"> • xs:unsignedByte • uint8_t_define • U8_define 				
Facets	<table> <tr> <td>maxInclusive</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><xs:simpleType name="U8_define"> <xs:annotation> <xs:documentation>8-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint8_t_define"/> </xs:simpleType></pre>				

Simple Type BYTE_define

Namespace	No namespace				
Annotations	byte type				
Diagram	<pre> graph LR BYTE_define -- restriction --> U8_define </pre>				
Type	U8_define				
Type hierarchy	<ul style="list-style-type: none"> • xs:unsignedByte • uint8_t_define • U8_define • BYTE_define 				
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><xs:simpleType name="BYTE_define"> <xs:annotation> <xs:documentation>byte type</xs:documentation> </xs:annotation> <xs:restriction base="U8_define"/> </xs:simpleType></pre>				

Simple Type I16_define

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

Diagram					
Type	int16_t_define				
Type hierarchy	<ul style="list-style-type: none"> xs:int <ul style="list-style-type: none"> int16_t_define <ul style="list-style-type: none"> I16_define 				
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><xs:simpleType name="I16_define"> <xs:restriction base="int16_t_define"/> </xs:simpleType></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"> xs:int <ul style="list-style-type: none"> uint16_t_define <ul style="list-style-type: none"> U16_define 				
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><xs:simpleType name="U16_define"> <xs:annotation> <xs:documentation>16-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint16_t_define"/> </xs:simpleType></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"> xs:integer <ul style="list-style-type: none"> int32_t_define <ul style="list-style-type: none"> I32_define 				
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><xs:simpleType name="I32_define"> <xs:annotation> <xs:documentation>32-bit signed integer</xs:documentation> </xs:annotation> <xs:restriction base="int32_t_define"/> </xs:simpleType></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"> xs:integer <ul style="list-style-type: none"> uint32_t_define <ul style="list-style-type: none"> U32_define 				
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><xs:simpleType name="U32_define"> <xs:annotation> <xs:documentation>16-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint32_t_define"/> </xs:simpleType></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"> xs:integer <ul style="list-style-type: none"> int64_t_define <ul style="list-style-type: none"> I64_define 				
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><xs:simpleType name="I64_define"> <xs:annotation> <xs:documentation>64-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="int64_t_define"/> </xs:simpleType></pre>				

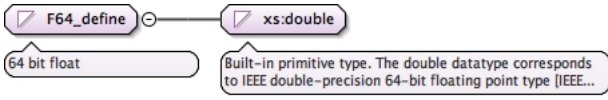
Simple Type U64_define

Namespace	No namespace				
Annotations	64-bit unsigned integer				
Diagram					
Type	uint64_t_define				
Type hierarchy	<ul style="list-style-type: none"> xs:integer <ul style="list-style-type: none"> uint64_t_define <ul style="list-style-type: none"> U64_define 				
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><xs:simpleType name="U64_define"> <xs:annotation> <xs:documentation>64-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint64_t_define"/> </xs:simpleType></pre>				

Simple Type F32_define

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

Simple Type F64_define

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

Attribute(s)

Attribute item / @name

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

Attribute item / @value

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

Attribute item / @comment

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

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

Attribute enum / @name

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

	<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></pre>
--	--

Attribute data_type_and_default_define / @default

Namespace	No namespace
Used by	Complex Type data_type_and_default_define
Source	<xs:attribute name="default" />

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	<xs:attribute name="size" type="positive_integer_define" />

Attribute parameter / @id

Namespace	No namespace
Annotations	ID of the attribute.
Type	No namespace
Properties	use: required
Facets	pattern ((0?x\d+) \d+)
Used by	Element parameter
Source	<xs:attribute name="id" use="required" type="id_define"> <xs:annotation base="id_define" type="id_define"> <xs:annot99.052001952 0 rg 1 0 0 -1 79.80000305 8.35200075200016.375 0.25 cm 0 1 -1 0 0 0 cm 0 0 m 0 -0.25 1 18. </xs:annot> </xs:annotation> </xs:attribute>

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

Attribute parameter / @name

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

Attribute parameters / @parameter_base

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

Attribute parameters / @opcode_base

Namespace	No namespace
Annotations	
Type	base_code_define
Properties	content: simple
Used by	Element parameters
Source	<pre><xs:attribute name="opcode_base" type="base_code_define"> <xs:annotation></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><xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute></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	<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	Element <code>arg_define/arg</code>						
Source	<pre><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></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><xs:attribute name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Comments about the argument.</xs:documentation> </xs:annotation> </xs:attribute></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><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"></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> <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> </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> <xs:attribute name="size" type="xs:nonNegativeInteger"> <xs:annotation> <xs:documentation>The size of the argument.</xs:documentation> </xs:annotation> </xs:attribute> </pre>

Attribute return / @name

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

Attribute return / @comment

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

Attribute external_arg_define / arg / @name

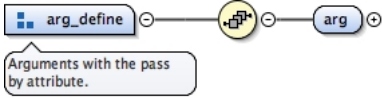
Namespace	No namespace
Annotations	Name of the argument.
Properties	use: required
Used by	Element external_arg_define/arg
Source	<pre><xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute></pre>

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><xs:attribute name="comment" type="xs:string"> <xs:annotation> <xs:documentation>Comments about the argument.</xs:documentation> </xs:annotation> </xs:attribute></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><xs:group name="arg_define"> <xs:annotation> <xs:documentation>Arguments with the pass by attribute.</xs:documentation> </xs:annotation></pre>

```
<xs:sequence>
  <xs:element name="arg">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="1" base="arg" >
```

```

        </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.		
Attributes	type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
Source	<pre> <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> </pre>			