Schema documentation for parameters_schema.xsd

august 31, 2016

Table of Contents

Namespace: ""

Schema(s)

Main schema parameters_schema.xsd

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

Included schema common_elements.xsd

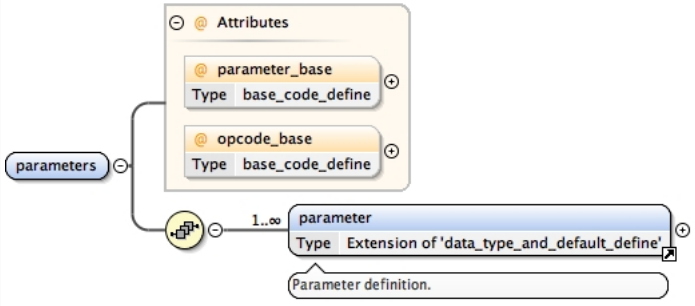
Namespace	No namespace
Properties	attribute form default: unqualified element form default: qualified

Included schema common_types.xsd

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

Element(s)

Element parameters

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

Instance	<parameters opcode_base="" parameter_base=""> <parameter data_type="" default="" id="" name="" save_opcode="" set_opcode="" size="">{1,unbounded}</parameter> </parameters>

Attributes

QName	Type	Use	
opcode_base	base_code_define	optional	
parameter_base	base_code_define	optional	

Source

```
<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>
```

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	<parameter data_type="" default="" id="" name="" save_opcode="" set_opcode="" size=""> <enum name="">{0,1}</enum> <comment>{1,1}</comment> </parameter>

Attributes

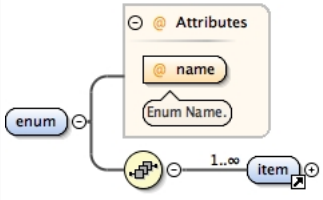
QName	Type	Use	
data_type	union of(not_user_cpp_type_define, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token,	required	

	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

```
<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>
```

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	`<enum name="">`
	`  <item comment="" name="" value="">{1,unbounded}</item>`
	`</enum>`

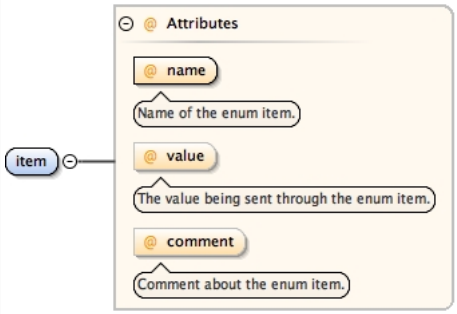
Attributes

QName	Type	Use	
name		required	
	Enum Name.		

Source

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

Element `item`

Namespace	No namespace
Diagram	

Properties	content:	complex
Used by	Element	enum

Attributes

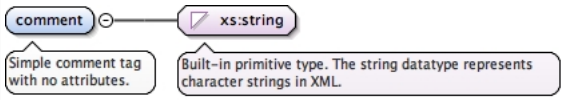
QName	Type	Use	
comment		optional	
	Comment about the enum item.		
name		required	
	Name of the enum item.		
value		optional	
	The value being sent through the enum item.		

Source

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

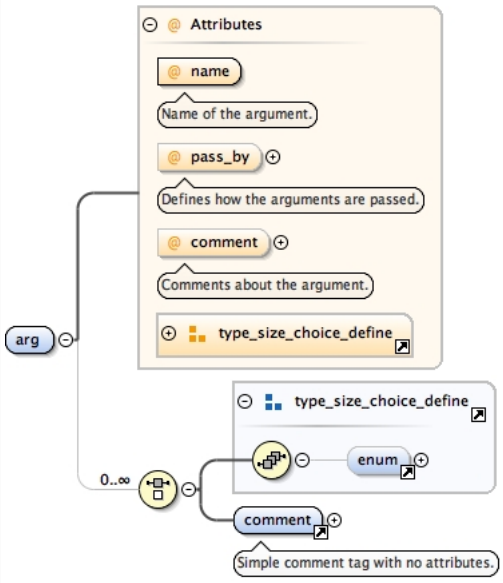
Element `comment`

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

Element `arg_define / arg`

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

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

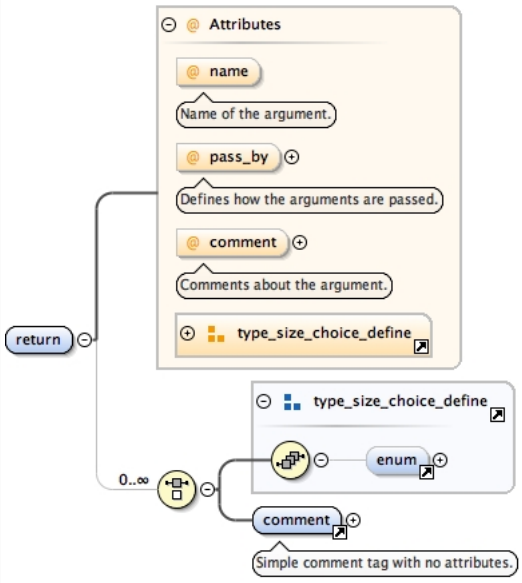
Attributes

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

Source

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

Element `return`

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

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

Attributes

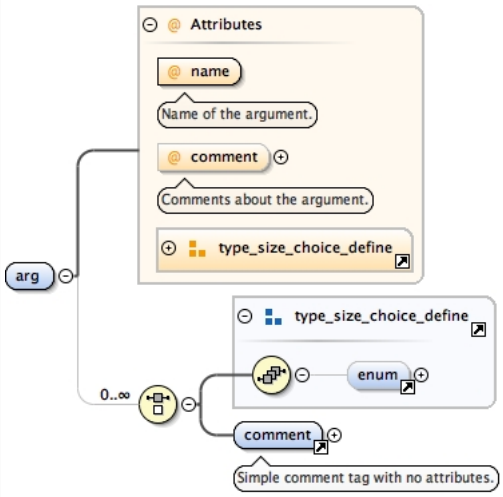
QName	Type	Use	
comment	xs:string	optional	
	Comments about the argument.		
data_type	union of(xs:string, restriction of xs:token, restriction of xs:token)	optional	
name		optional	
	Name of the argument.		
pass_by	pass_by_define	optional	
	Defines how the arguments are passed.		
size	xs:nonNegativeInteger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

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

Source

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

Element `external_arg_define / arg`

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

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

Attributes

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

	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

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

Complex Type(s)

Complex Type `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

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, restriction of xs:token, restriction of xs:token, restriction of xs:token, restriction of xs:token)	required	
default		optional	
size	positive_integer_define	optional	

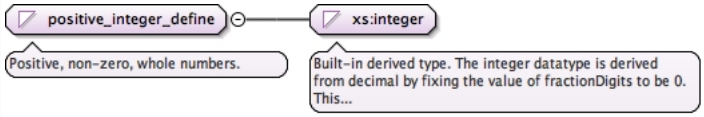
Source

```
<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: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>
```

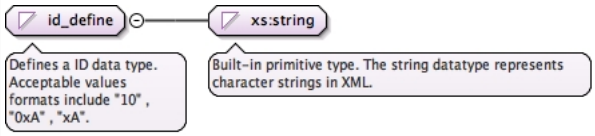
Simple Type(s)

Simple Type `positive_integer_define`

Namespace	No namespace
Annotations	`Positive, non-zero, whole numbers.`
Diagram	
Type	restriction of xs:integer
Facets	minInclusive `1`
Used by	Attribute data_type_and_default_define/@size

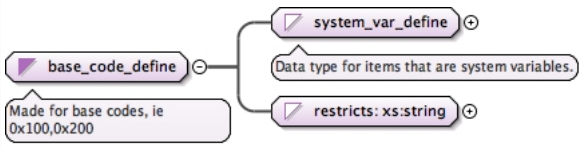
Source	`<xs:simpleType name="positive_integer_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Positive, non-zero, whole numbers.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:integer">`
	`  <xs:minInclusive value="1"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `id_define`

Namespace	No namespace
Annotations	`Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".`
Diagram	
Type	restriction of xs:string
Facets	pattern `((0?x\d+)|\d+)`
Used by	Attributes parameter/@id, parameter/@save_opcode, parameter/@set_opcode

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

Simple Type `base_code_define`

Namespace	No namespace
Annotations	`Made for base codes, ie 0x100,0x200`
Diagram	
Type	union of(system_var_define, restriction of xs:string)
Used by	Attributes parameters/@opcode_base, parameters/@parameter_base

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

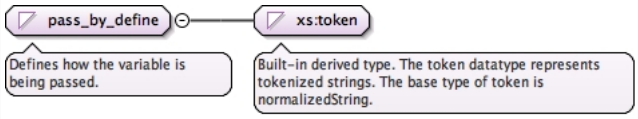
```
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

Simple Type `full_items_define`

Namespace	No namespace
Annotations	`Valid values for the full tag.`
Diagram	
Type	restriction of xs:token
Facets	enumeration `drop` enumeration `assert` enumeration `block`

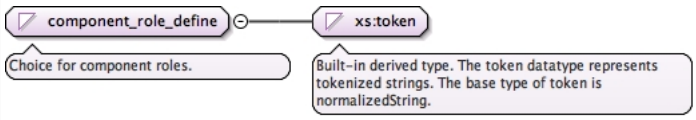
Source	`<xs:simpleType name="full_items_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Valid values for the full tag.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:token">`
	`  <xs:enumeration value="drop"/>`
	`  <xs:enumeration value="assert"/>`
	`  <xs:enumeration value="block"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `pass_by_define`

Namespace	No namespace
Annotations	`Defines how the variable is being passed.`
Diagram	
Type	restriction of xs:token
Facets	enumeration `reference` enumeration `value` enumeration `pointer`
Used by	Attributes arg_define/arg/@pass_by, return/@pass_by

Source	`<xs:simpleType name="pass_by_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines how the variable is being passed.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:token">`
	`  <xs:enumeration value="reference"/>`
	`  <xs:enumeration value="value"/>`
	`  <xs:enumeration value="pointer"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `component_role_define`

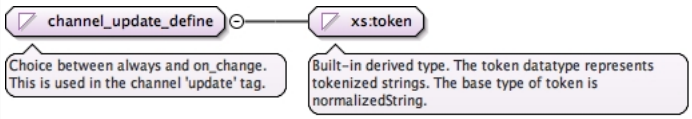
Namespace	No namespace
Annotations	`Choice for component roles.`
Diagram	
Type	restriction of xs:token
Facets	enumeration `LogEvent`

	enumeration	LogTextEvent
	enumeration	TimeGet
	enumeration	ParamSet
	enumeration	ParamGet
	enumeration	Telemetry
	enumeration	CmdRegistration
	enumeration	Cmd
	enumeration	CmdResponse

Source

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

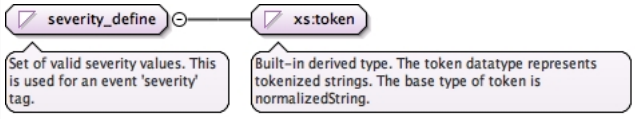
Simple Type channel_update_define

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

Source

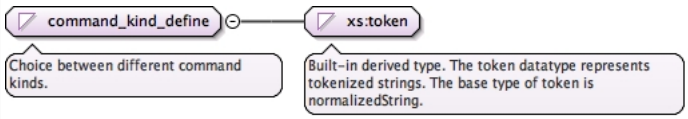
```
<xs:simpleType name="channel_update_define">
  <xs:annotation>
    <xs:documentation>Choice between always and on_change. This is used in the channel 'update'
 tag.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="always"/>
    <xs:enumeration value="on_change"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type severity_define

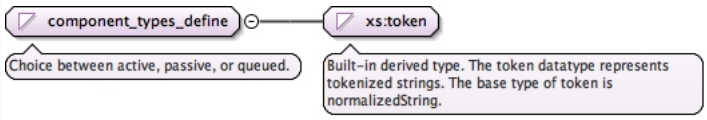
Namespace	No namespace	
Annotations	Set of valid severity values. This is used for an event 'severity' tag.	
Diagram		
Type	restriction of xs:token	
Facets	enumeration	COMMAND
	enumeration	ACTIVITY_LO
	enumeration	ACTIVITY_HI
	enumeration	WARNING_LO
	enumeration	WARNING_HI
	enumeration	DIAGNOSTIC

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

Simple Type command_kind_define

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

Simple Type component_types_define

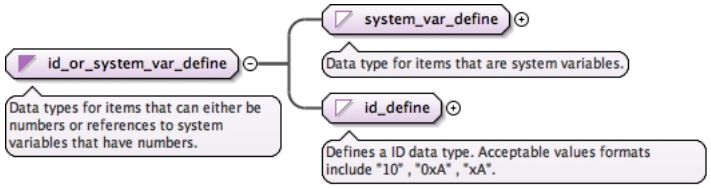
Namespace	No namespace
Annotations	Choice between active, passive, or queued.
Diagram	
Type	restriction of xs:token
Facets	enumeration active enumeration passive enumeration queued
Source	<xs:simpleType name="component_types_define"> <xs:annotation> <xs:documentation>Choice between active, passive, or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

Simple Type port_types_define

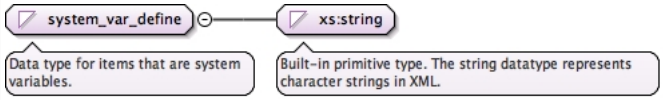
Namespace	No namespace

Annotations	Choice between different port types.
Diagram	
Type	restriction of xs:token
Facets	enumeration input enumeration sync_input enumeration guarded_input enumeration async_input enumeration model_input enumeration output
Source	<xs:simpleType name="port_types_define"> <xs:annotation> <xs:documentation>Choice between different port types.</xs:documentation> </xs:annotation> <xs:restriction base="xs:token"> <xs:enumeration value="input"/> <xs:enumeration value="sync_input"/> <xs:enumeration value="guarded_input"/> <xs:enumeration value="async_input"/> <xs:enumeration value="model_input"/> <xs:enumeration value="output"/> </xs:restriction> </xs:simpleType>

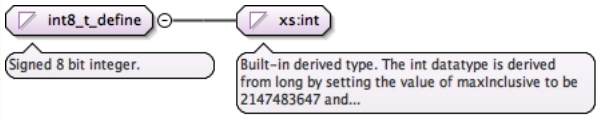
Simple Type id_or_system_var_define

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

Simple Type system_var_define

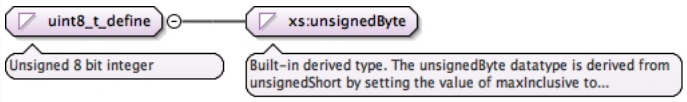
Namespace	No namespace
Annotations	Data type for items that are system variables.
Diagram	
Type	restriction of xs:string
Facets	pattern $[\w|_|\-]+
Source	<xs:simpleType name="system_var_define"> <xs:annotation> <xs:documentation>Data type for items that are system variables.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:pattern value="$[\w|_|\-]+"/> </xs:restriction> </xs:simpleType>

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

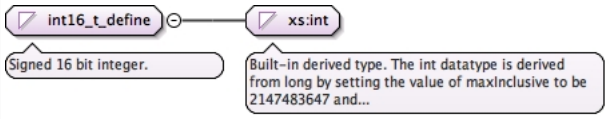
Source	`<xs:simpleType name="int8_t_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Signed 8 bit integer.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:int">`
	`  <xs:minInclusive value="-128"/>`
	`  <xs:maxInclusive value="127"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `uint8_t_define`

Namespace	No namespace
Annotations	`Unsigned 8 bit integer`
Diagram	
Type	restriction of xs:unsignedByte
Facets	maxInclusive `255` minInclusive `0`
Used by	Simple Type U8_define

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

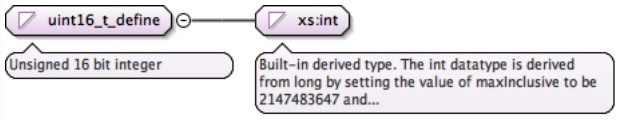
Simple Type `int16_t_define`

Namespace	No namespace
Annotations	`Signed 16 bit integer.`
Diagram	
Type	restriction of xs:int
Facets	maxInclusive `32767` minInclusive `-32768`
Used by	Simple Type I16_define

Source	`<xs:simpleType name="int16_t_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Signed 16 bit integer.</xs:documentation>`
	`  </xs:annotation>`

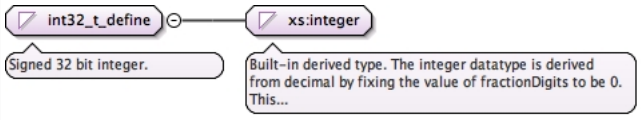
```
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-32768"/>
    <xs:maxInclusive value="32767"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `uint16_t_define`

Namespace	No namespace
Annotations	`Unsigned 16 bit integer`
Diagram	
Type	restriction of xs:int
Facets	maxInclusive `65535` minInclusive `0`
Used by	Simple Type U16_define
Source	

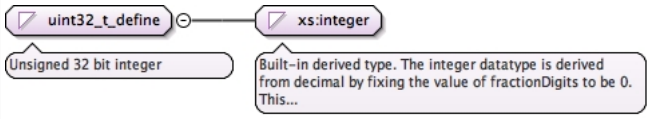
```
<xs:simpleType name="uint16_t_define">
  <xs:annotation>
    <xs:documentation>Unsigned 16 bit integer</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="65535"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `int32_t_define`

Namespace	No namespace
Annotations	`Signed 32 bit integer.`
Diagram	
Type	restriction of xs:integer
Facets	maxInclusive `2147483647` minInclusive `-2147483648`
Used by	Simple Types I32_define, NATIVE_INT_TYPE_define
Source	

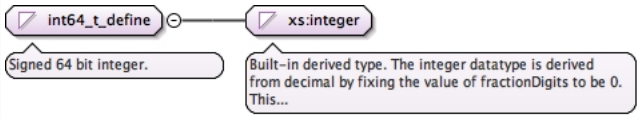
```
<xs:simpleType name="int32_t_define">
  <xs:annotation>
    <xs:documentation>Signed 32 bit integer.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="-2147483648"/>
    <xs:maxInclusive value="2147483647"/>
  </xs:restriction>
</xs:simpleType>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

Simple Type `uint32_t_define`

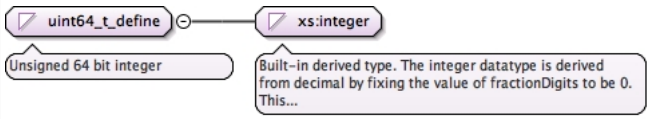
Namespace	No namespace
Annotations	`Unsigned 32 bit integer`
Diagram	
Type	restriction of xs:integer
Facets	maxInclusive `4294967295` minInclusive `0`

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

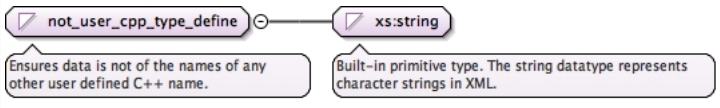
Simple Type `int64_t_define`

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

Simple Type `uint64_t_define`

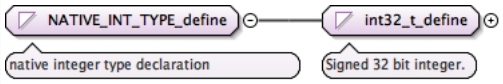
Namespace	No namespace	
Annotations	Unsigned 64 bit integer	
Diagram		
Type	restriction of xs:integer	
Facets	maxInclusive	18446744073709551615
	minInclusive	0
Used by	Simple Type	U64_define
Source	<xs:simpleType name="uint64_t_define"> <xs:annotation> <xs:documentation>Unsigned 64 bit integer</xs:documentation>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

Simple Type `not_user_cpp_type_define`

Namespace	No namespace
Annotations	Ensures data is not of the names of any other user defined C++ name.
Diagram	

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

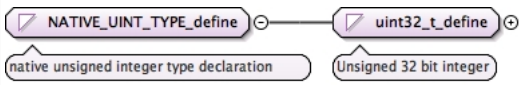
Simple Type `NATIVE_INT_TYPE_define`

Namespace	No namespace
Annotations	native integer type declaration
Diagram	
Type	int32_t_define

Type hierarchy	• xs:integer
	  • int32_t_define
	  • NATIVE_INT_TYPE_define

Facets	maxInclusive 2147483647 minInclusive -2147483648
Source	<xs:simpleType name="NATIVE_INT_TYPE_define"> <xs:annotation> <xs:documentation>native integer type declaration</xs:documentation> </xs:annotation> <xs:restriction base="int32_t_define"/> </xs:simpleType>

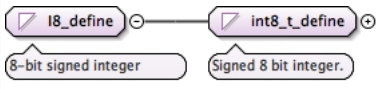
Simple Type `NATIVE_UINT_TYPE_define`

Namespace	No namespace
Annotations	native unsigned integer type declaration
Diagram	
Type	uint32_t_define

Type hierarchy	• xs:integer
	  • uint32_t_define
	  • NATIVE_UINT_TYPE_define

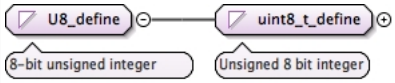
Facets	maxInclusive 4294967295 minInclusive 0
Source	<xs:simpleType name="NATIVE_UINT_TYPE_define"> <xs:annotation> <xs:documentation>native unsigned integer type declaration</xs:documentation> </xs:annotation> <xs:restriction base="uint32_t_define"/> </xs:simpleType>

Simple Type `I8_define`

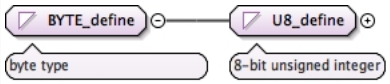
Namespace	No namespace
Annotations	8-bit signed integer
Diagram	
Type	int8_t_define
Type hierarchy	• xs:int

	• int8_t_define • I8_define
Facets	maxInclusive 127 minInclusive -128
Source	<xs:simpleType name="I8_define"> <xs:annotation> <xs:documentation>8-bit signed integer</xs:documentation> </xs:annotation> <xs:restriction base="int8_t_define"/> </xs:simpleType>

Simple Type U8_define

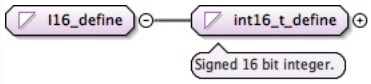
Namespace	No namespace
Annotations	8-bit unsigned integer
Diagram	
Type	uint8_t_define
Type hierarchy	• xs:unsignedByte • uint8_t_define • U8_define
Facets	maxInclusive 255 minInclusive 0
Used by	Simple Type BYTE_define
Source	<xs:simpleType name="U8_define"> <xs:annotation> <xs:documentation>8-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint8_t_define"/> </xs:simpleType>

Simple Type BYTE_define

Namespace	No namespace
Annotations	byte type
Diagram	
Type	U8_define
Type hierarchy	• xs:unsignedByte • uint8_t_define • U8_define • BYTE_define
Facets	maxInclusive 255 minInclusive 0
Source	<xs:simpleType name="BYTE_define"> <xs:annotation> <xs:documentation>byte type</xs:documentation> </xs:annotation> <xs:restriction base="U8_define"/> </xs:simpleType>

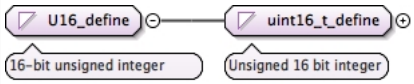
Simple Type I16_define

Namespace	No namespace

Diagram	
Type	int16_t_define
Type hierarchy	• xs:int • int16_t_define • I16_define
Facets	maxInclusive 32767 minInclusive -32768

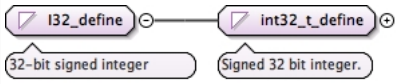
Source	`<xs:simpleType name="I16_define">`
	`  <xs:restriction base="int16_t_define"/>`
	`</xs:simpleType>`

Simple Type `U16_define`

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

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

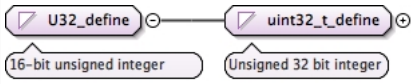
Simple Type `I32_define`

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

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

Simple Type `U32_define`

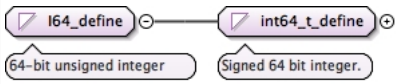
Namespace	No namespace

Annotations	16-bit unsigned integer
Diagram	
Type	uint32_t_define

Type hierarchy	• xs:integer
	  • uint32_t_define
	  • U32_define

Facets	maxInclusive 4294967295 minInclusive 0
Source	<xs:simpleType name="U32_define"> <xs:annotation> <xs:documentation>16-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint32_t_define"/> </xs:simpleType>

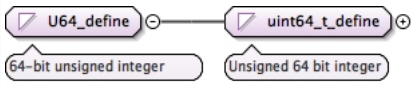
Simple Type I64_define

Namespace	No namespace
Annotations	64-bit unsigned integer
Diagram	
Type	int64_t_define

Type hierarchy	• xs:integer
	  • int64_t_define
	  • I64_define

Facets	maxInclusive 9223372036854775807 minInclusive -9223372036854775808
Source	<xs:simpleType name="I64_define"> <xs:annotation> <xs:documentation>64-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="int64_t_define"/> </xs:simpleType>

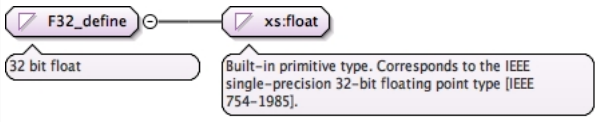
Simple Type U64_define

Namespace	No namespace
Annotations	64-bit unsigned integer
Diagram	
Type	uint64_t_define

Type hierarchy	• xs:integer
	  • uint64_t_define
	  • U64_define

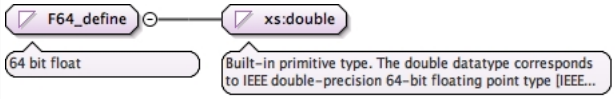
Facets	maxInclusive 18446744073709551615 minInclusive 0
Source	<xs:simpleType name="U64_define"> <xs:annotation> <xs:documentation>64-bit unsigned integer</xs:documentation> </xs:annotation> <xs:restriction base="uint64_t_define"/> </xs:simpleType>

Simple Type `F32_define`

Namespace	No namespace
Annotations	`32 bit float`
Diagram	
Type	xs:float

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

Simple Type `F64_define`

Namespace	No namespace
Annotations	`64 bit float`
Diagram	
Type	xs:double

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

Attribute(s)

Attribute `item / @name`

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

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

Attribute `item / @value`

Namespace	No namespace
Annotations	`The value being sent through the enum item.`
Used by	Element item

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

Attribute `item / @comment`

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

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

Attribute `enum / @name`

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

Attribute `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, 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	<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>
```

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	id_define
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>`
	`  <xs:documentation>ID of the attribute.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

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	<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>	

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	<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>	

Attribute `parameter / @name`

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

Attribute `parameters / @parameter_base`

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

Attribute `parameters / @opcode_base`

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

```
    <xs:documentation/>
  </xs:annotation>
</xs:attribute>
```

Attribute `arg_define / arg / @name`

Namespace	No namespace
Annotations	`Name of the argument.`
Properties	use: `required`
Used by	Element arg_define/arg

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

Attribute `arg_define / arg / @pass_by`

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

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

Attribute `arg_define / arg / @comment`

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

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

Attribute `type_size_choice_define / @data_type`

Namespace	No namespace
Type	union of(xs:string, restriction of xs:token, restriction of xs:token)
Properties	content: `simple`
Used by	Attribute Group type_size_choice_define

Source	`<xs:attribute name="data_type">`
	`  <xs:simpleType>`
	`  <xs:union memberTypes="xs:string">`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:token">`
	`  <xs:enumeration value="string"/>`
	`  </xs:restriction>`
	`  </xs:simpleType>`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:token">`

```
          <xs:enumeration value="ENUM"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</xs:attribute>
```

Attribute `type_size_choice_define / @type`

Namespace	No namespace
Type	union of(xs:string, restriction of xs:token, restriction of xs:token)
Properties	content: `simple`
Used by	Attribute Group type_size_choice_define
Source	

```
<xs:attribute name="type">
  <xs:simpleType>
    <xs:union memberTypes="xs:string">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="string"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="ENUM"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</xs:attribute>
```

Attribute `type_size_choice_define / @size`

Namespace	No namespace
Annotations	`The size of the argument.`
Type	xs:nonNegativeInteger
Properties	content: `simple`
Used by	Attribute Group type_size_choice_define
Source	

```
<xs:attribute name="size" type="xs:nonNegativeInteger">
  <xs:annotation>
    <xs:documentation>The size of the argument.</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute `return / @name`

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

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

Attribute `return / @pass_by`

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

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

Attribute `return / @comment`

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

Attribute `external_arg_define / arg / @name`

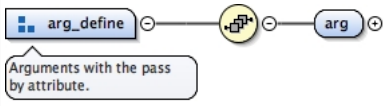
Namespace	No namespace
Annotations	Name of the argument.
Properties	use: required
Used by	Element external_arg_define/arg
Source	<xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the argument.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `external_arg_define / arg / @comment`

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

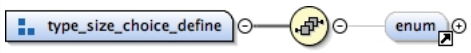
Element Group(s)

Element Group `arg_define`

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

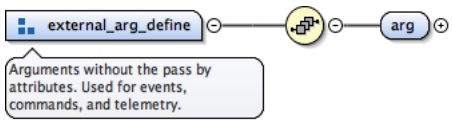
```
  <xs:sequence>
    <xs:element name="arg">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:group ref="type_size_choice_define"/>
          <xs:element ref="comment"/>
        </xs:choice>
        <xs:attribute name="name" use="required">
          <xs:annotation>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

Element Group `type_size_choice_define`

Namespace	No namespace
Diagram	
Used by	Elements arg_define/arg, external_arg_define/arg, return
Model	enum{0,1}
Children	enum

Source	`<xs:group name="type_size_choice_define">`
	`  <xs:sequence>`
	`  <xs:element minOccurs="0" ref="enum"/>`
	`  </xs:sequence>`
	`</xs:group>`

Element Group `external_arg_define`

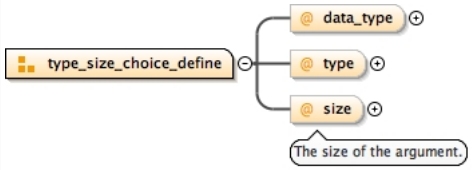
Namespace	No namespace
Annotations	Arguments without the pass by attributes. Used for events, commands, and telemetry.
Diagram	
Model	arg
Children	arg

Source	`<xs:group name="external_arg_define">`
	`  <xs:annotation>`
	`  <xs:documentation>Arguments without the pass by attributes. Used for events, commands, and telemetry.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="arg">`
	`  <xs:complexType>`
	`  <xs:choice minOccurs="0" maxOccurs="unbounded">`
	`  <xs:group ref="type_size_choice_define"/>`
	`  <xs:element ref="comment"/>`
	`  </xs:choice>`
	`  <xs:attribute name="name" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation>Name of the argument.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:attribute>`
	`  <xs:attribute name="comment" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>Comments about the argument.</xs:documentation>`

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

Attribute Group(s)

Attribute Group `type_size_choice_define`

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

Attributes

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

Source

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