

Schema documentation for channel_schema.xsd

august 31, 2016

Table of Contents

```
channel_schema.xsd
common_elements.xsd
common_types.xsd
```

```
telemetry
channel
enum
item
comment
arg_define / arg
return
external_arg_define / arg
```

```
id_define
channel_update_define
base_code_define
full_items_define
pass_by_define
component_role_define
severity_define
command_kind_define
component_types_define
port_types_define
id_or_system_var_define
system_var_define
positive_integer_define
int8_t_define
uint8_t_define
int16_t_define
uint16_t_define
int32_t_define
uint32_t_define
int64_t_define
uint64_t_define
not_user_cpp_type_define
NATIVE_INT_TYPE_define
NATIVE_UINT_TYPE_define
I8_define
U8_define
BYTE_define
I16_define
U16_define
I32_define
U32_define
I64_define
U64_define
F32_define
F64_define
```

```
item / @name
item / @value
item / @comment
enum / @name
channel / @id
channel / @name
channel / @update
channel / @abbrev
channel / @format_string
```

```
channel / @high_yellow
channel / @high_red
channel / @high_orange
channel / @low_yellow
channel / @low_red
channel / @low_orange
type_size_choice_define / @data_type
type_size_choice_define / @type
type_size_choice_define / @size
telemetry / @telemetry_base
arg_define / arg / @name
arg_define / arg / @pass_by
arg_define / arg / @comment
return / @name
return / @pass_by
return / @comment
external_arg_define / arg / @name
external_arg_define / arg / @comment

type_size_choice_define
arg_define
external_arg_define

type_size_choice_define
```

Namespace: ""

Schema(s)

Main schema channel_schema.xsd

	unqualified
	qualified

Included schema common_elements.xsd

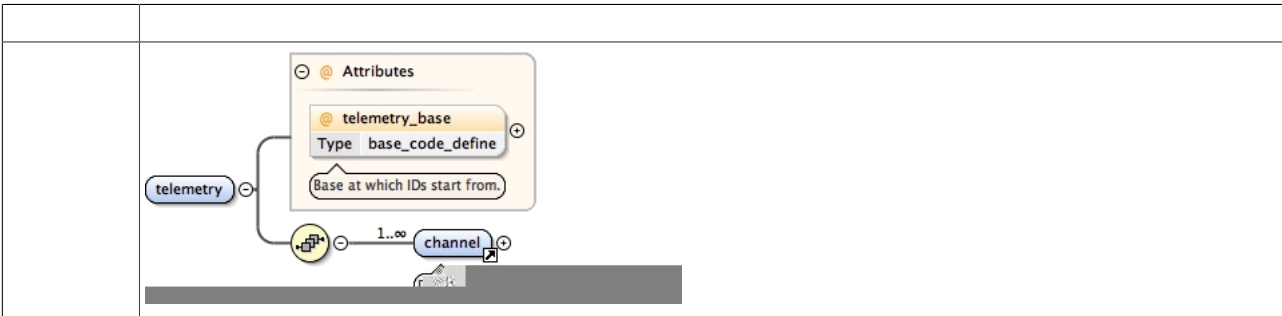
	unqualified
	qualified

Included schema common_types.xsd

	unqualified
	qualified

Element(s)

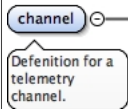
Element telemetry



	complex			
	<pre><telemetry telemetry_base=""> <channel abbrev="" data_type="" format_string="" high_orange="" high_red="" high_yellow="" id="" low channel> </telemetry></pre>			
	QName	Type	Use	
	telemetry_base			
		Base at which IDs start from.		
	<pre><xs:element name="telemetry"> <xs:complexType> <xs:sequence> <xs:element maxOccurs="unbounded" ref="channel"/> </xs:sequence> <xs:attribute name="telemetry_base" type="base_code_define"> <xs:annotation> <xs:documentation>Base at which IDs start from.</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType> </xs:element></pre>			

Element channel

	Defenition for a telemetry channel.



```
<channel abbrev="" data_type="" format_string="" high_orange="" high_red="" high_yellow="" id="" low_orange="" low
  <enum name="">{0,1}</enum>
  <comment>{1,1}</comment>
</channel>
```

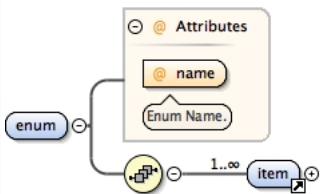
QName	Type	Use	
abbrev			
	Required for AMPCS dictionary generation.		
data_type			
format_string			
	Used to format data into readable content on the ground system software.		
high_orange			

QName	Type	Use	
high_red			
high_yellow			
id			
	ID for the telemetry channel.		
low_orange			
low_red			
low_yellow			
name			
	Name of the telemetry channel.		
size			
	The size of the argument.		
type			
update			
	Defines when the channel updates.		

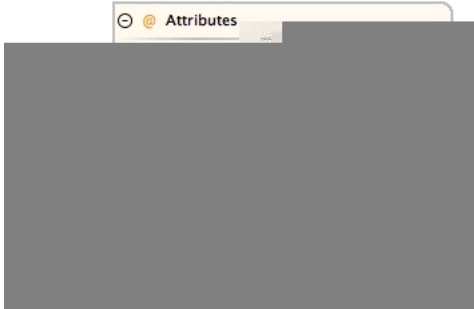
<pre><xs:element name="channel"> <xs:annotation> <xs:documentation>Defenition for a telemetry channel.</xs:documentation> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:group ref="type_size_choice_define"/> <xs:element ref="comment"/> </xs:choice> <xs:attribute name="id" use="required" type="id_define"> <xs:annotation> <xs:documentation>ID for the telemetry channel.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the telemetry channel.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="update" type="channel_update_define"> <xs:annotation> <xs:documentation>Defines when the channel updates.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="abbrev"> <xs:annotation> <xs:documentation>Required for AMPCS dictionary generation.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="format_string"> <xs:annotation> <xs:documentation>Used to format data into readable content on the ground system software.</ xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="high_yellow" type="xs:decimal"/> <xs:attribute name="high_red" type="xs:decimal"/> <xs:attribute name="high_orange" type="xs:decimal"/> <xs:attribute name="low_yellow" type="xs:decimal"/> <xs:attribute name="low_red" type="xs:decimal"/> <xs:attribute name="low_orange" type="xs:decimal"/> <xs:attributeGroup ref="type_size_choice_define"/> </xs:complexType> </xs:element></pre>

Element enum

--	--

													
	complex												
	<pre><enum name=" "> <item comment=" " name=" " value=" ">{1,unbounded}</item> </enum></pre>												
	<table><tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr><tr><td>name</td><td></td><td></td><td></td></tr><tr><td></td><td colspan="3">Enum Name.</td></tr></table>	QName	Type	Use		name					Enum Name.		
QName	Type	Use											
name													
	Enum Name.												
	<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

																													
	complex																												
	<table><tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr><tr><td>comment</td><td></td><td></td><td></td></tr><tr><td></td><td colspan="3">Comment about the enum item.</td></tr><tr><td>name</td><td></td><td></td><td></td></tr><tr><td></td><td colspan="3">Name of the enum item.</td></tr><tr><td>value</td><td></td><td></td><td></td></tr><tr><td></td><td colspan="3">The value being sent through the enum item.</td></tr></table>	QName	Type	Use		comment					Comment about the enum item.			name					Name of the enum item.			value					The value being sent through the enum item.		
QName	Type	Use																											
comment																													
	Comment about the enum item.																												
name																													
	Name of the enum item.																												
value																													
	The value being sent through the enum item.																												
	<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"></pre>																												

	<pre> <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

	Simple comment tag with no attributes.
	<div> <div>comment</div> <div>xs:string</div> <div>Simple comment tag with no attributes.</div> <div>Built-in primitive type. The string datatype represents character strings in XML.</div> </div>
	simple
	<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

	complex												
	<pre><arg comment="" data_type="" name="" pass_by="" size="" type=""> <enum name="">{0,1}</enum> <comment>{1,1}</comment> </arg></pre>												
	<table><tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr><tr><td>comment</td><td></td><td></td><td></td></tr><tr><td></td><td colspan="3">Comments about the argument.</td></tr></table>	QName	Type	Use		comment					Comments about the argument.		
QName	Type	Use											
comment													
	Comments about the argument.												

QName	Type	Use	
data_type			
name			
	Name of the argument.		
pass_by			
	Defines how the arguments are passed.		
size			
	The size of the argument.		
type			


```

<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>

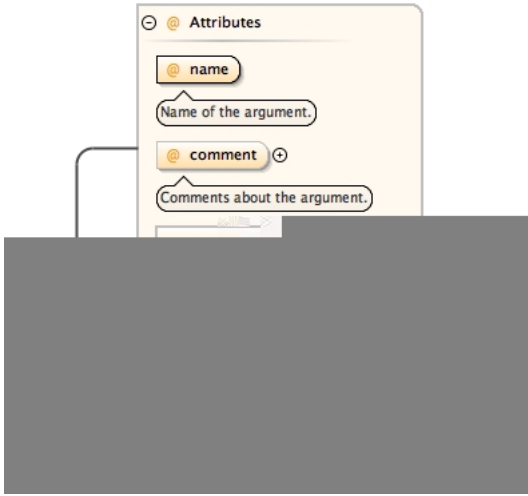
```

Element return

	complex
	<pre> <return comment="" data_type="" name="" pass_by="" size="" type=""> </pre>

	<pre><enum name="">{0,1}</enum> <comment>{1,1}</comment> </return></pre>		
	QName	Type	Use
	comment		
		Comments about the argument.	
	data_type		
	name		
		Name of the argument.	
	pass_by		
		Defines how the arguments are passed.	
	size		
		The size of the argument.	
	type		
	<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>		

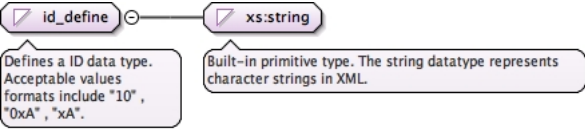
Element external_arg_define / arg

			
	complex		

	<pre><arg comment="" data_type="" name="" size="" type=""> <enum name="">{0,1}</enum> <comment>{1,1}</comment> </arg></pre>			
	QName	Type	Use	
	comment			
		Comments about the argument.		
	data_type			
	name			
		Name of the argument.		
	size			
		The size of the argument.		
	type			
	<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>			

Simple Type(s)

Simple Type id_define

	Defines a ID data type. Acceptable values formats include "10" , "0xA" , "xA".	
		
	((0?x\d+) \d+)	
	<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 channel_update_define

--	--

	Choice between always and on_change. This is used in the channel 'update' tag.
	<p>Choice between always and on_change. This is used in the channel 'update' tag.</p> <p>Built-in derived type. The token datatype represents</p>
	<div>always</div> <div>on_change</div>
	<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 base_code_define

	Made for base codes, ie 0x100,0x200
	<p>Made for base codes, ie 0x100,0x200</p> <p>system_var_define</p> <p>Data type for items that are system variables.</p> <p>restricts: xs:string</p>
	<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>

Simple Type full_items_define

	Valid values for the full tag.
	<p>Valid values for the full tag.</p> <p>Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.</p>
	<div>drop</div> <div>assert</div> <div>block</div>
	<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" /> </pre>

```

    <xs:enumeration value="assert" />
    <xs:enumeration value="block" />
  </xs:restriction>
</xs:simpleType>

```

Simple Type pass_by_define

Defines how the variable is being passed.



reference
value
pointer

```

<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

Choice for component roles.

LogEvent
LogTextEvent
TimeGet
ParamSet
ParamGet
Telemetry
CmdRegistration
Cmd
CmdResponse

```

<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 severity_define

	Set of valid severity values. This is used for an event 'severity' tag.
	COMMAND
	ACTIVITY_LO
	ACTIVITY_HI
	WARNING_LO
	WARNING_HI
	DIAGNOSTIC
	FATAL
	<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

	Choice between different command kinds.
	async
	sync
	guarded
	<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

	Choice between active, passive, or queued.

	<div> <div>component types de</div> <div></div> </div>
	<div>active</div> <div>passive</div> <div>queued</div>
	<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

	Choice between different port types.
	<div> <div>port_types_define</div> <div>xs.token</div> <div>Choice between different port types.</div> <div>Built-in derived type. The token datatype represents</div> </div>
	<div>input</div> <div>sync_input</div> <div>guarded_input</div> <div>async_input</div> <div>model_input</div> <div>output</div>
	<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

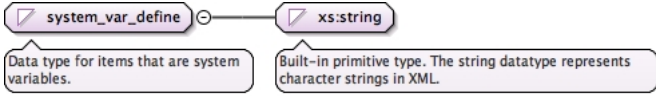
	Data types for items that can either be numbers or references to system variables that have numbers.
	<div> <div>id_or_system_var_define</div> <div>system_var_define</div> <div>Data type for items that are system variables.</div> <div>id_define</div> <div>Data types for items that can either be numbers or references to system variables that have numbers.</div> <div>Define</div> </div>
	<pre> <xs:simpleType name="id_or_system_var_define"> <xs:annotation> </pre>

```

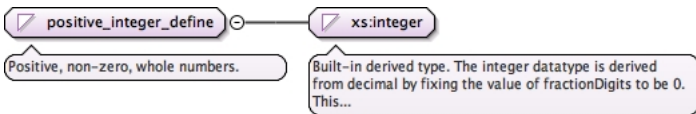
    <xs:documentation>Data types for items that can either be numbers or references to system
    variables that have numbers.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="system_var_define id_define"/>
</xs:simpleType>

```

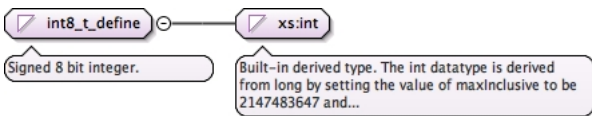
Simple Type `system_var_define`

	Data type for items that are system variables.
	
	$\$[\backslash w _ \\-]+$
	<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 `positive_integer_define`

	Positive, non-zero, whole numbers.
	
	1
	<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 `int8_t_define`

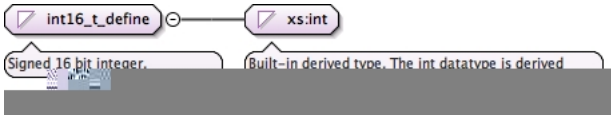
	Signed 8 bit integer.
	
	127
	-128
	<pre> <xs:simpleType name="int8_t_define"> <xs:annotation> <xs:documentation>Signed 8 bit integer.</xs:documentation> </xs:annotation> <xs:restriction base="xs:int"> <xs:minInclusive value="-128"/> <xs:maxInclusive value="127"/> </xs:restriction> </xs:simpleType> 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

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

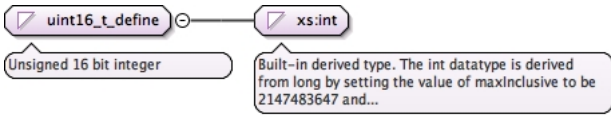
Simple Type uint8_t_define

	Unsigned 8 bit integer
	
	255
	0
	<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

	Signed 16 bit integer.
	
	32767
	-32768
	<pre><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></pre>

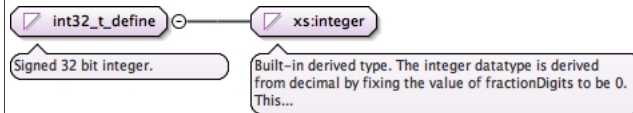
Simple Type uint16_t_define

	Unsigned 16 bit integer
	
	65535
	0
	<pre><xs:simpleType name="uint16_t_define"></pre>

<pre> <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

	Signed 32 bit integer.
--	------------------------

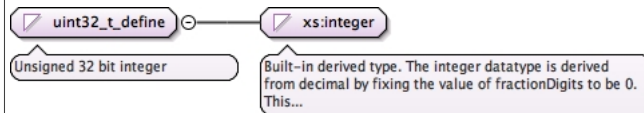


	2147483647
	-2147483648

<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

	Unsigned 32 bit integer
--	-------------------------

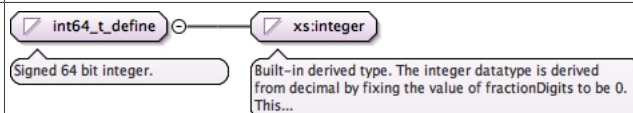


	4294967295
	0

<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>
--

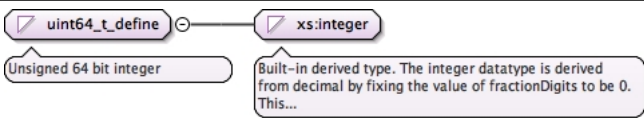
Simple Type int64_t_define

	Signed 64 bit integer.
--	------------------------

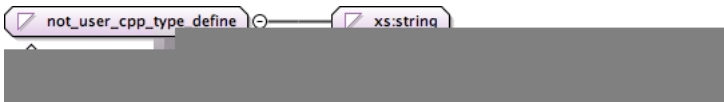


	9223372036854775807
	-9223372036854775808
	<pre><xs:simpleType name="int64_t_define"> <xs:annotation> <xs:documentation>Signed 64 bit integer.</xs:documentation> </xs:annotation> <xs:restriction base="xs:integer"> <xs:minInclusive value="-9223372036854775808"/> <xs:maxInclusive value="9223372036854775807"/>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


Simple Type uint64_t_define

	Unsigned 64 bit integer
	
	18446744073709551615
	0
	<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

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

Simple Type NATIVE_INT_TYPE_define

	native integer type declaration
	

	<div>2147483647</div> <div>-2147483648</div>
	<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

	native unsigned integer type declaration
	
	<div>4294967295</div> <div>0</div>
	<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

	8-bit signed integer
	
	<div>127</div> <div>-128</div>
	<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

	8-bit unsigned integer
	

	<div>255</div> <div>0</div>
	<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

	byte type
	<div>255</div> <div>0</div>
	<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

	<div>32767</div> <div>-32768</div>
	<pre><xs:simpleType name="I16_define"> <xs:restriction base="int16_t_define"/> </xs:simpleType></pre>

Simple Type U16_define

--	--

	16-bit unsigned integer
	
	<div>65535</div> <hr/> <div>0</div>
	<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

	32-bit signed integer
	
	<div>2147483647</div> <hr/> <div>-2147483648</div>
	<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

	16-bit unsigned integer
	
	<div>4294967295</div> <hr/> <div>0</div>
	<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

	64-bit unsigned integer
	
	<div>9223372036854775807</div> <div>-9223372036854775808</div>
	<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

	64-bit unsigned integer
	
	<div>18446744073709551615</div> <div>0</div>
	<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

	32 bit float
	
	<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

--	--

	64 bit float
	<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

	Name of the enum item.
	required
	<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

	The value being sent through the enum item.
	<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

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

Attribute enum / @name

	Enum Name.
	required
	<pre><xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Enum Name.</xs:documentation> </xs:annotation> </xs:attribute></pre>

Attribute channel / @id

--	--

	ID for the telemetry channel.
	required
	((0?x\d+) \d+)
	<pre><xs:attribute name="id" use="required" type="id_define"> <xs:annotation> <xs:documentation>ID for the telemetry channel.</xs:documentation> </xs:annotation> </xs:attribute></pre>

Attribute channel / @name

	Name of the telemetry channel.
	required
	<pre><xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>Name of the telemetry channel.</xs:documentation> </xs:annotation> </xs:attribute></pre>

Attribute channel / @update

	Defines when the channel updates.
	simple
	always
	on_change
	<pre><xs:attribute name="update" type="channel_update_define"> <xs:annotation> <xs:documentation>Defines when the channel updates.</xs:documentation> </xs:annotation> </xs:attribute></pre>

Attribute channel / @abbrev

	Required for AMPCS dictionary generation.
	<pre><xs:attribute name="abbrev"> <xs:annotation> <xs:documentation>Required for AMPCS dictionary generation.</xs:documentation> </xs:annotation> </xs:attribute></pre>

Attribute channel / @format_string

	Used to format data into readable content on the ground system software.
	<pre><xs:attribute name="format_string"> <xs:annotation> <xs:documentation>Used to format data into readable content on the ground system software.</xs:documentation> </xs:annotation> </xs:attribute></pre>

	</xs:attribute>
--	-----------------

Attribute channel / @high_yellow

	simple
	<xs:attribute name="high_yellow" type="xs:decimal"/>

Attribute channel / @high_red

	simple
	<xs:attribute name="high_red" type="xs:decimal"/>

Attribute channel / @high_orange

	simple
	<xs:attribute name="high_orange" type="xs:decimal"/>

Attribute channel / @low_yellow

	simple
	<xs:attribute name="low_yellow" type="xs:decimal"/>

Attribute channel / @low_red

	simple
	<xs:attribute name="low_red" type="xs:decimal"/>

Attribute channel / @low_orange

	simple
	<xs:attribute name="low_orange" type="xs:decimal"/>

Attribute type_size_choice_define / @data_type

--	--

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

Attribute type_size_choice_define / @type

	simple
	<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

	The size of the argument.
	simple
	<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 telemetry / @telemetry_base

	Base at which IDs start from.
	simple
	<pre> <xs:attribute name="telemetry_base" type="base_code_define"> </pre>

	<pre> <xs:annotation> <xs:documentation>Base at which IDs start from.</xs:documentation> </xs:annotation> </xs:attribute> </pre>
--	--

Attribute arg_define / arg / @name

	Name of the argument.
	required
	<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

	Defines how the arguments are passed.
	simple
	reference
	value
	pointer
	<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

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

Attribute return / @name

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

Attribute return / @pass_by

--	--

	Defines how the arguments are passed.
	simple
	reference
	value
	pointer
	<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

	Comments about the argument.
	simple
	<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

	Name of the argument.
	required
	<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

	Comments about the argument.
	simple
	<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 type_size_choice_define

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

Element Group arg_define

	Arguments with the pass by attribute.
	<p>Arguments with the pass by attribute.</p>
	<pre> <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> </pre>

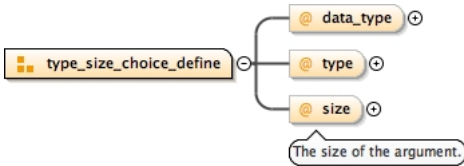
Element Group external_arg_define

	Arguments without the pass by attributes. Used for events, commands, and telemetry.
	<p>Arguments without the pass by attributes. Used for events, commands, and telemetry.</p>
	<pre> <xs:group name="external_arg_define"> <xs:annotation> <xs:documentation>Arguments without the pass by attributes. Used for events, commands, and telemetry.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="arg"> <xs:complexType> </pre>

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

Attribute Group(s)

Attribute Group type_size_choice_define

																								
	<table><tr><th>QName</th><th>Type</th><th>Use</th><th></th></tr><tr><td>data_type</td><td></td><td></td><td></td></tr><tr><td>size</td><td></td><td></td><td></td></tr><tr><td></td><td colspan="3">The size of the argument.</td></tr><tr><td>type</td><td></td><td></td><td></td></tr></table>	QName	Type	Use		data_type				size					The size of the argument.			type						
QName	Type	Use																						
data_type																								
size																								
	The size of the argument.																							
type																								
	<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:attributeGroup></pre>																							

```
</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>
```
