Isode

Red/Black Application Profiles for use with XML Guard

24th January 2024

1 Overview

This white paper specifies two Application Profiles to support use with Isode's Red/Black product with an XML Guard. It defines:

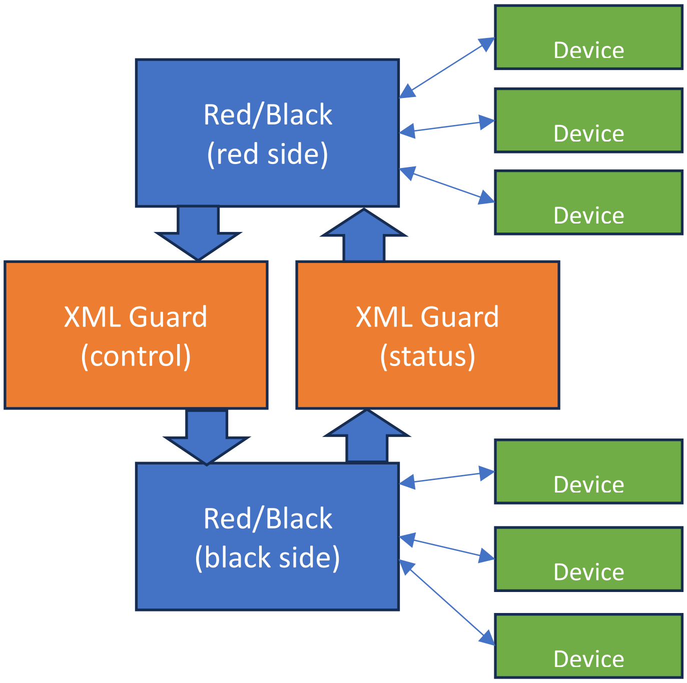
- Schemas for each Application Profile
- Normalization Requirements
- Rules which can be used to constrain the base profiles

Application Profiles are summarized in the Isode white paper "XML Guard Application Profiles". They define a product-independent approach to specify functionality across an XML Guard.

Goals of this specification:

- To provide a clear specification to enable system accreditation of a system using Isode's M-Guard and Isode's Red/Black products.
- To enable use of a third party XML Guard with Isode's Red/Black product.

2 The Red/Black XML Protocols



Red/Black is a monitoring and control product that supports HF communications chain management. It supports operation over a red/black boundary, so that red side operators can monitor and control black side devices. Red/Black servers operate as a pair, with one red side and one black side server.

The Red/Black servers communicate using Isode's open GCXP (Guard Content Exchange Protocol) with a set of XML messages specified in this document. This is a control protocol flowing from red to black, which enables red side operator control of the black side devices. There is a status protocol flowing from black to red, which enables red side monitoring of the black side devices.

The core of the control protocol is a <control/> message with three string parameters and complex value that is used to set a single parameter on a black side device. The elements are:

1. Device Name. A name that identifies the specific device, such as "Radio 1" or "Antenna 3".
2. Device Family. This identifies the type of device and is associated with an abstract specification of the device family which contains a list of status and control parameters supported by the device.
3. Parameter. This string is from the device abstract specification and identifies the parameter that is being modified.
4. Parameter value. A range of encodings, shared with the status protocol, which allow a wide range of parameter types to be encoded.

The core of the status protocol is a <status/> message that has the same parameters as the <control/> message plus the option to include a device-specific <alert/> message to enable flexible alerting from the device.

This communication can use M-Guard or another XML Guard to provide red/black separation.

3 Summary of the Profiles

There are two profiles, one associated with the status flow and the other associated with the control flow.

These profiles are based on XML schemas, which defines the XML messages exchanged and are specified below. The Schemas, specified as an XML Schema Definition (XSD), specify an "outer bound" for what is allowed through the guard.

The associated rules then constrain the schemas, by blocking elements of the schema. This allows the messages being passed to be further constrained.

4 Normalization

Both profiles require the following message normalization:

- Prohibition of XML Comments and XML Processing Instructions.
- Use of Canonical XML. Following Canonical XML Version 1.1 of May 2008.
- Unicode Normalization following UNICODE NORMALIZATION FORMS 13.0.0 using Normalization Form C (NFC) "Canonical Decomposition, followed by Canonical Composition"

5 Isode Red/Black Application Profile Products

Isode provides two Application Profile products that follow the two profiles defined in his specification. These profiles enable M-Guard to provide guards compliant to these profiles.

6 Red/Black Status Profile

6.1 Black to Red Protection Requirements

The primary security requirement for black to red information flow is to prevent malware or other attacks. The Red/Black status messages are highly structured XML and this structure is an effective way to prevent malware.

There are some parts of the protocol which allow larger data elements. Rules are defined that enable these to be blocked or constrained.

6.2 Rules

This version of the Application Profile defines the following associated rules, that may be enabled to further constrain the base schema. The choice of rules will be deployment-specific.

Rule	Notes
Prohibit Photos	The messages enable JPEG Photos to be communicated, which could support visual monitoring. This rule blocks such messages. It is recommended to enable this rule unless photos are being used.
String Size Limit	Strings are used in various places in the protocol. No limits are defined in the base schema, as specific applications may have need for long strings. This rule enables a generic maximum string size to be set, in order to prevent very large strings to be sent. It is recommended to set this to a value suitable for the deployment.

6.3 Schema of Status Protocol

The Red/Black Status protocol schema is specified below. It defines the <status/> message sent from black to red.

6.3.1 Example Messages

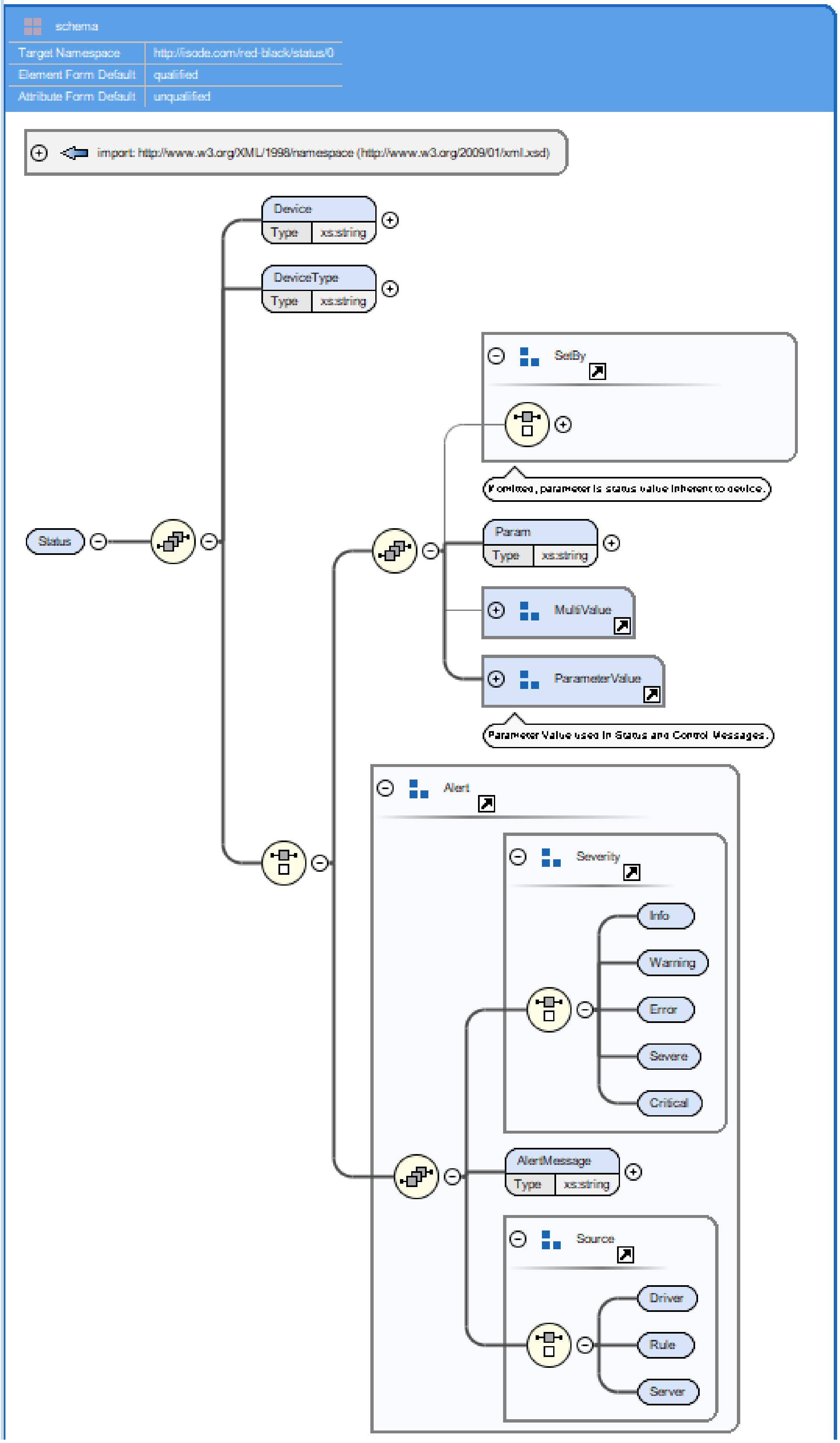
This section shows some sample messages, correctly normalized following this specification, but folded to make easier to read.

```
<Status xmlns="http://isode.com/red-black/status/0">
<Device>HF2000</Device>
<DeviceType>Radio</DeviceType>
<Param>Status</Param><
Enumerated>Operational</Enumerated></Status>

<Status xmlns="http://isode.com/red-black/status/0">
<Device>Local Host</Device>
<DeviceType>Host</DeviceType>
<Param>HrSystemProcesses</Param>
<Integer>227</Integer></Status>

<Status xmlns="http://isode.com/red-black/status/0">
<Device>IESMATrIX</Device>
<DeviceType>IESMATRIXRFHF4x4</DeviceType>
<Critical></Critical><AlertMessage>read failure</AlertMessage>
<Server></Server></Status>
```

6.3.2 Schema Visualisation

This section shows the schema, as visualised by the oXygen XML editor.

This is the overall schema showing structure of the messages



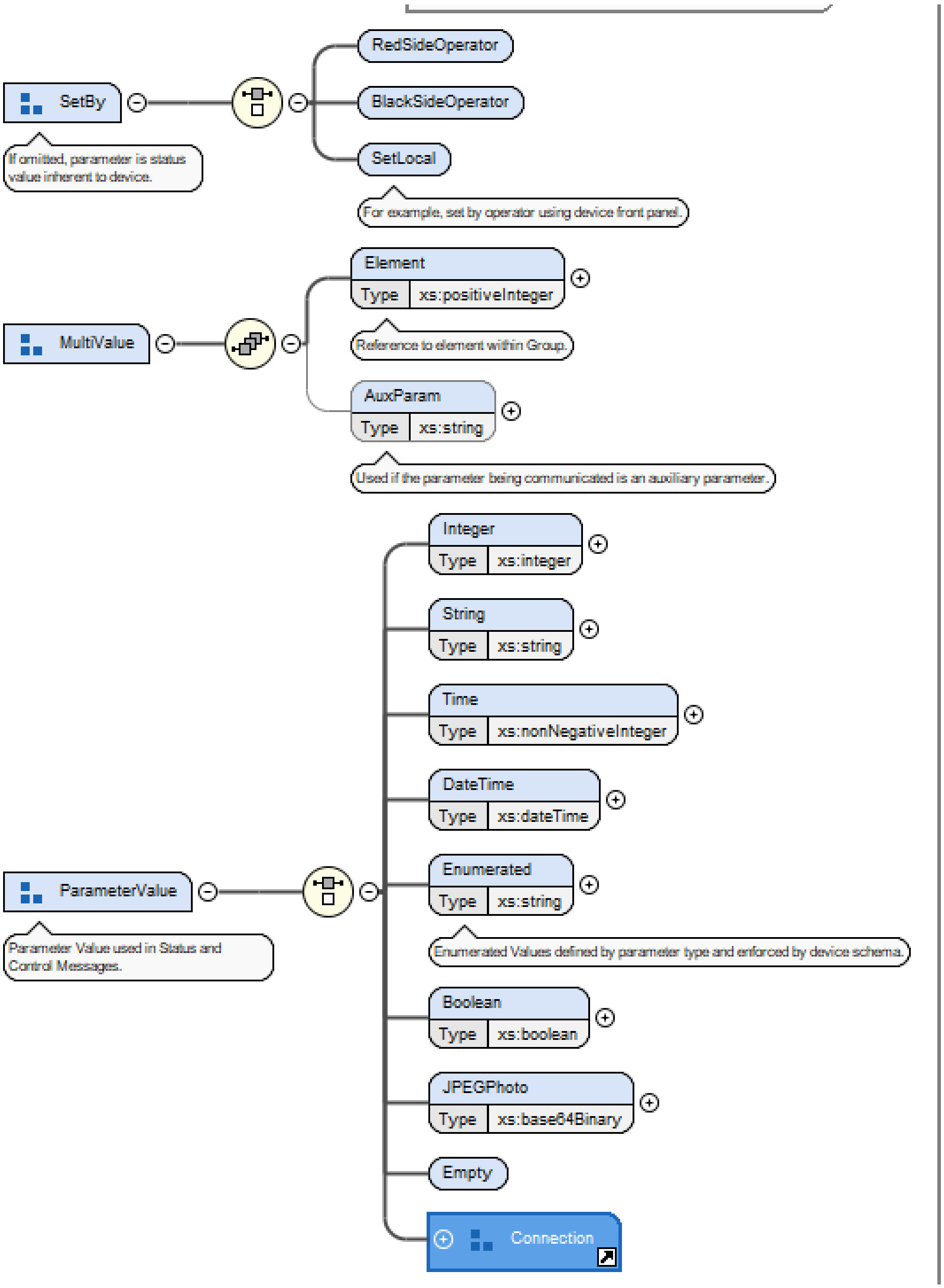
schema

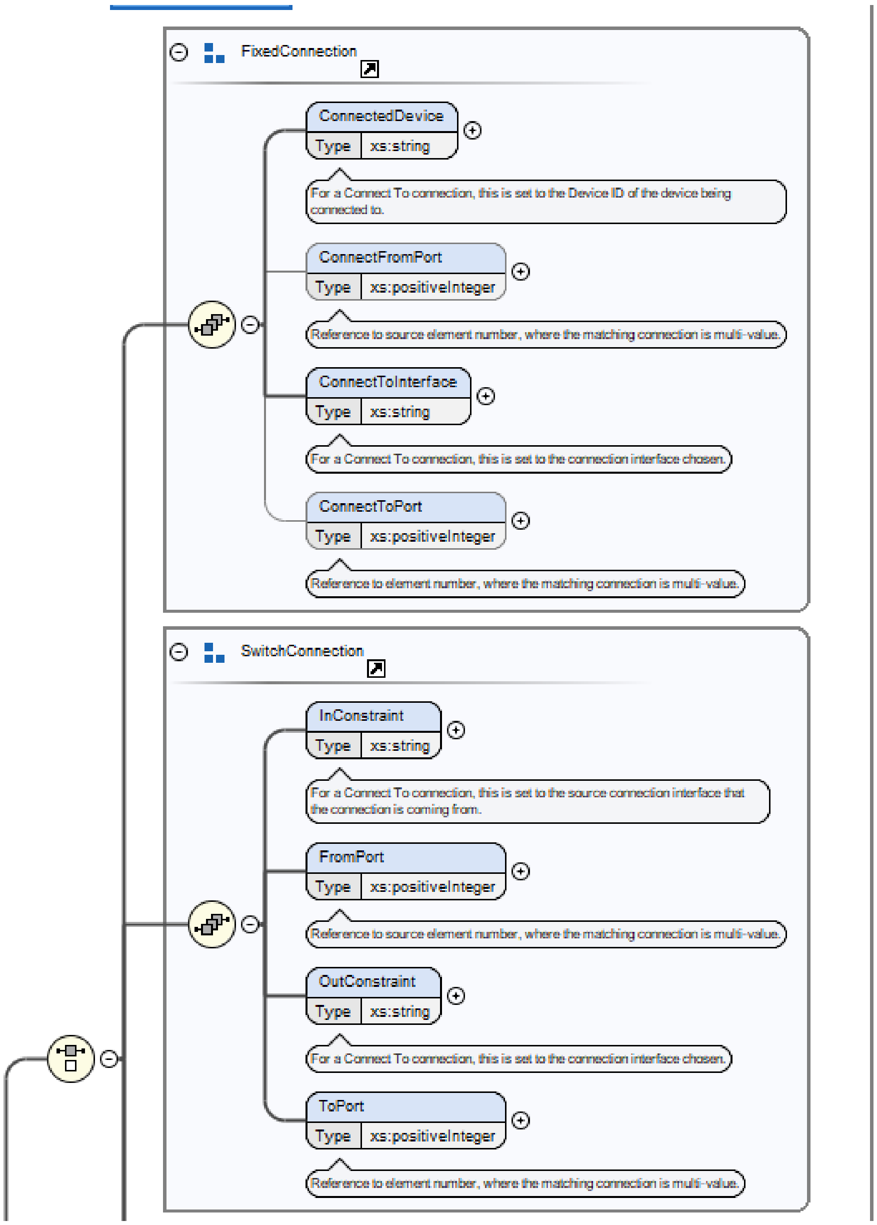
Target Namespace	http://isode.com/red-black/status/0
Element Form Default	qualified
Attribute Form Default	unqualified

import: http://www.w3.org/XML/1998/namespace (http://www.w3.org/2009/01/xml.xsd)

If omitted, parameter is status value inherent to device.

Parameter Value used in Status and Control Messages.


SetBy
If omitted, parameter is status value inherent to device.
RedSideOperator
BlackSideOperator
SetLocal
For example, set by operator using device front panel.
MultiValue
Element
Type xs:positiveInteger
Reference to element within Group.
AuxParam
Type xs:string
Used if the parameter being communicated is an auxiliary parameter.
ParameterValue
Parameter Value used in Status and Control Messages.
Integer
Type xs:integer
String
Type xs:string
Time
Type xs:nonNegativeInteger
DateTime
Type xs:dateTime
Enumerated
Type xs:string
Enumerated Values defined by parameter type and enforced by device schema.
Boolean
Type xs:boolean
JPEGPhoto
Type xs:base64Binary
Empty
Connection



For a Connect To connection, this is set to the Device ID of the device being connected to.

Reference to source element number, where the matching connection is multi-value.

For a Connect To connection, this is set to the connection interface chosen.

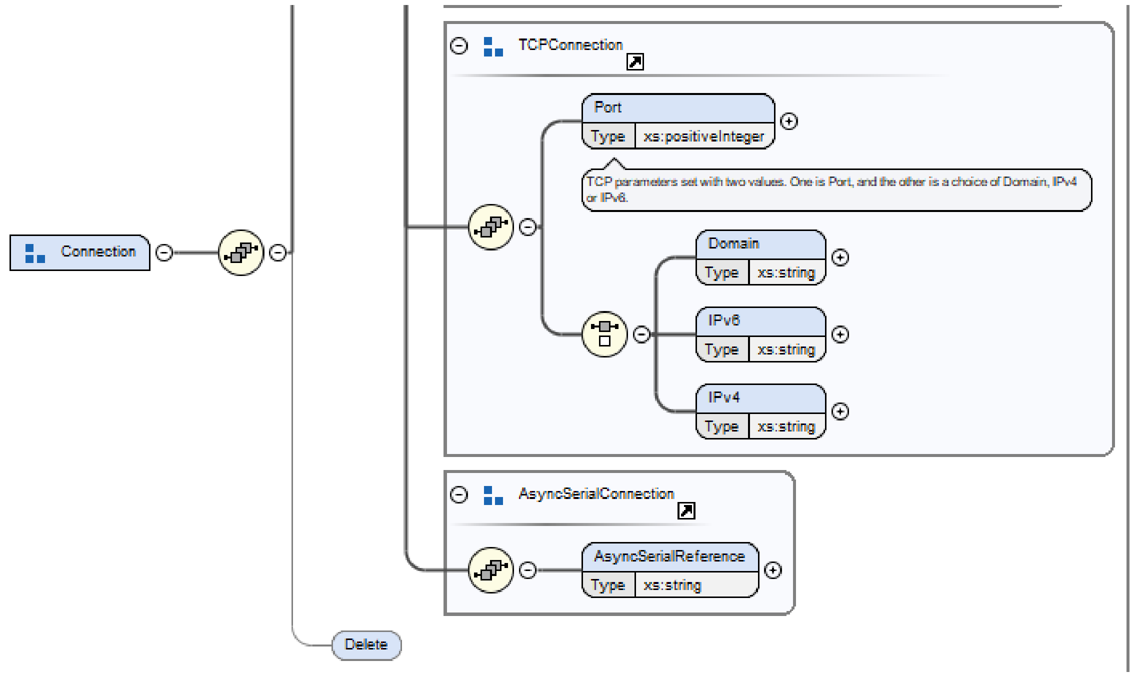
Reference to element number, where the matching connection is multi-value.

For a Connect To connection, this is set to the source connection interface that the connection is coming from.

Reference to source element number, where the matching connection is multi-value.

For a Connect To connection, this is set to the connection interface chosen.

Reference to element number, where the matching connection is multi-value.



6.3.3 Schema Specification

This is the formal XML Schema Definition:

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://isode.com/red-black/status/0" xml:lang="en" targetNamespace="http://isode.com/red-black/status/0" elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>

    <xs:element name="Status">

        <xs:annotation>

            <xs:documentation/>

        </xs:annotation>

        <xs:complexType>

            <xs:sequence>

                <xs:element name="Device" type="xs:string">

                    <xs:annotation>

                        <xs:documentation/>

                    </xs:annotation>

                </xs:element>

                <xs:element name="DeviceType" type="xs:string"/>

                <xs:choice>

                    <xs:sequence>

                        <xs:group ref="SetBy" minOccurs="0"/>

                        <xs:element name="Param" type="xs:string">
```

```
                            <xs:annotation>
                                <xs:documentation/>
                            </xs:annotation>
                        </xs:element>
                        <xs:group ref="MultiValue" minOccurs="0"/>
                        <xs:group ref="ParameterValue"/>
                    </xs:sequence>
                    <xs:group ref="Alert"/>
                </xs:choice>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:group name="SetBy">
        <xs:annotation>
            <xs:documentation>If omitted, parameter is status value inherent to device.
      </xs:documentation>
        </xs:annotation>
        <xs:choice>
            <xs:element name="RedSideOperator">
                <xs:complexType/>
            </xs:element>
            <xs:element name="BlackSideOperator">
                <xs:complexType/>
            </xs:element>
            <xs:element name="SetLocal">
                <xs:annotation>
                    <xs:documentation>For example, set by operator using device front
panel.
          </xs:documentation>
                </xs:annotation>
                <xs:complexType/>
            </xs:element>
        </xs:choice>
    </xs:group>
    <xs:group name="MultiValue">
        <xs:sequence>
            <xs:element name="Element" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference to element within Group.
          </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="AuxParam" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Used if the parameter being communicated is an
            auxiliary parameter.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="ParameterValue">
        <xs:annotation>
            <xs:documentation>Parameter Value used in Status and Control Messages.
      </xs:documentation>
        </xs:annotation>
        <xs:choice>
            <xs:element name="Integer" type="xs:integer"/>
            <xs:element name="String" type="xs:string"/>
            <xs:element name="Time" type="xs:nonNegativeInteger"/>
            <xs:element name="DateTime" type="xs:dateTime"/>
            <xs:element name="Enumerated" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Enumerated Values defined by parameter type and
             enforced by device schema.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Boolean" type="xs:boolean"/>
            <xs:element name="JPEGPhoto" type="xs:base64Binary"/>
            <xs:element name="Empty">
                <xs:complexType/>
            </xs:element>
            <xs:group ref="Connection"/>
        </xs:choice>
    </xs:group>
    <xs:group name="Connection">
        <xs:sequence>
            <xs:choice>
                <xs:group ref="FixedConnection"/>
                <xs:group ref="SwitchConnection"/>
                <xs:group ref="TCPConnection"/>
                <xs:group ref="AsyncSerialConnection"/>
            </xs:choice>
            <xs:element name="Delete" minOccurs="0">
                <xs:complexType/>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="SwitchConnection">
        <xs:sequence>
            <xs:element name="InConstraint" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the source connection interface that
the
            connection is coming from.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="FromPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to source element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="OutConstraint" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the connection interface
chosen.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ToPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="FixedConnection">
        <xs:sequence>
            <xs:element name="ConnectedDevice" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the Device ID of the device being
connected to. </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="ConnectFromPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to source element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ConnectToInterface" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the connection interface
chosen.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="ConnectToPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="TCPConnection">
        <xs:sequence>
            <xs:element name="Port" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>TCP
            parameters set with two values. One is Port, and the other is a choice of
Domain, IPv4 or
            IPv6.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:choice>
                <xs:element name="Domain" type="xs:string"/>
                <xs:element name="IPv6" type="xs:string"/>
                <xs:element name="IPv4" type="xs:string"/>
            </xs:choice>
        </xs:sequence>
    </xs:group>
    <xs:group name="AsyncSerialConnection">
        <xs:sequence>
            <xs:element name="AsyncSerialReference" type="xs:string"/>
        </xs:sequence>
    </xs:group>
    <xs:group name="Alert">
        <xs:sequence>
            <xs:group ref="Severity"/>
            <xs:element name="AlertMessage" type="xs:string"/>
            <xs:group ref="Source"/>
        </xs:sequence>
    </xs:group>
    <xs:group name="Severity">
        <xs:choice>
            <xs:element name="Info">
                <xs:complexType/>
            </xs:element>
            <xs:element name="Warning">
                <xs:complexType/>
            </xs:element>
            <xs:element name="Error">
                <xs:complexType/>
            </xs:element>
            <xs:element name="Severe">
                <xs:complexType/>
            </xs:element>
            <xs:element name="Critical">
                <xs:complexType/>
            </xs:element>
        </xs:choice>
    </xs:group>
    <xs:group name="Source">
        <xs:choice>
            <xs:element name="Driver">
                <xs:complexType/>
            </xs:element>
            <xs:element name="Rule">
                <xs:complexType/>
            </xs:element>
            <xs:element name="Server">
                <xs:complexType/>
            </xs:element>
        </xs:choice>
    </xs:group>
```

```
</xs:schema>
```

7 Red/Black Control Profile

7.1 Red to Black Protection Requirements

A key security requirement is to prevent leakage of sensitive information from Red to Black. The minimal information contained in the control messages supports this.

Another security concern may be covert channel, where extraneous information in the protocol is used to convey other information. M-Guard Rate Control is a key tool to minimize this risk.

Rules as part of this application can constrain the information further, in order to reduce potential for covert signaling.

It is anticipated that the status protocol will send many attributes from all devices with a wide range of encodings. Control over the red/black boundary will be much more restricted. Only selected devices will be controlled; only selected attributes on those devices will be changed. Generally attributes modified will have a tight syntax (typically integer) and often only a constrained set of values are allowed. Rules are specified to constrain the protocol in this way.

7.2 Rules

This version of the Application Profile defines the following associated rules, that may be enabled to further constrain the base schema. These rules are set out to broadly correspond to the schema order, but grouping related functions together. The choice of rules will be deployment-specific.

The broad goal of rules is to restrict control messages to things that it is deemed necessary to control.

The Rule Catalog is generated from the Red/Black configuration, as rules have knowledge of Red/Black option.

Rule	Notes
Allowed Device Types	This rule allows setting a list of allowed device types. Messages with any other device type will be blocked. If this rule is not set, any device type is allowed.
Allowed Device Names	There is a rule for each device type (e.g., Radio). A rule for a devices type has a list of parameters which are named devices (e.g., “Radio 1”) that are valid for devices of this type. When this rule is enabled, messages with the device type will be blocked unless the device name matches one of the listed devices.

Rule	Notes
Valid Parameter	This single rule will enforce that the parameter type in a message is one that is allowed for the device type. This rule also enforces that the correct parameter encoding (e.g, Integer) is used for the parameter. It is anticipated that this rule will usually be selected.
Allowed Parameters	There is a rule for each device type, which specifies the parameter types that are allowed for this device type. If this rule is enabled, only the parameter types specified are allowed.
Allowed Values	There is a rule for each parameter type/device type pair, which can be configured with a list of valid values (integers or strings). If a rule for a parameter type is enabled, this parameter is restricted to the specified values.

7.3 Schema of Control Protocol

The Red/Black control protocol schema is specified below. It specifies the <control/> messages sent from red to black.

7.3.1 Example Messages

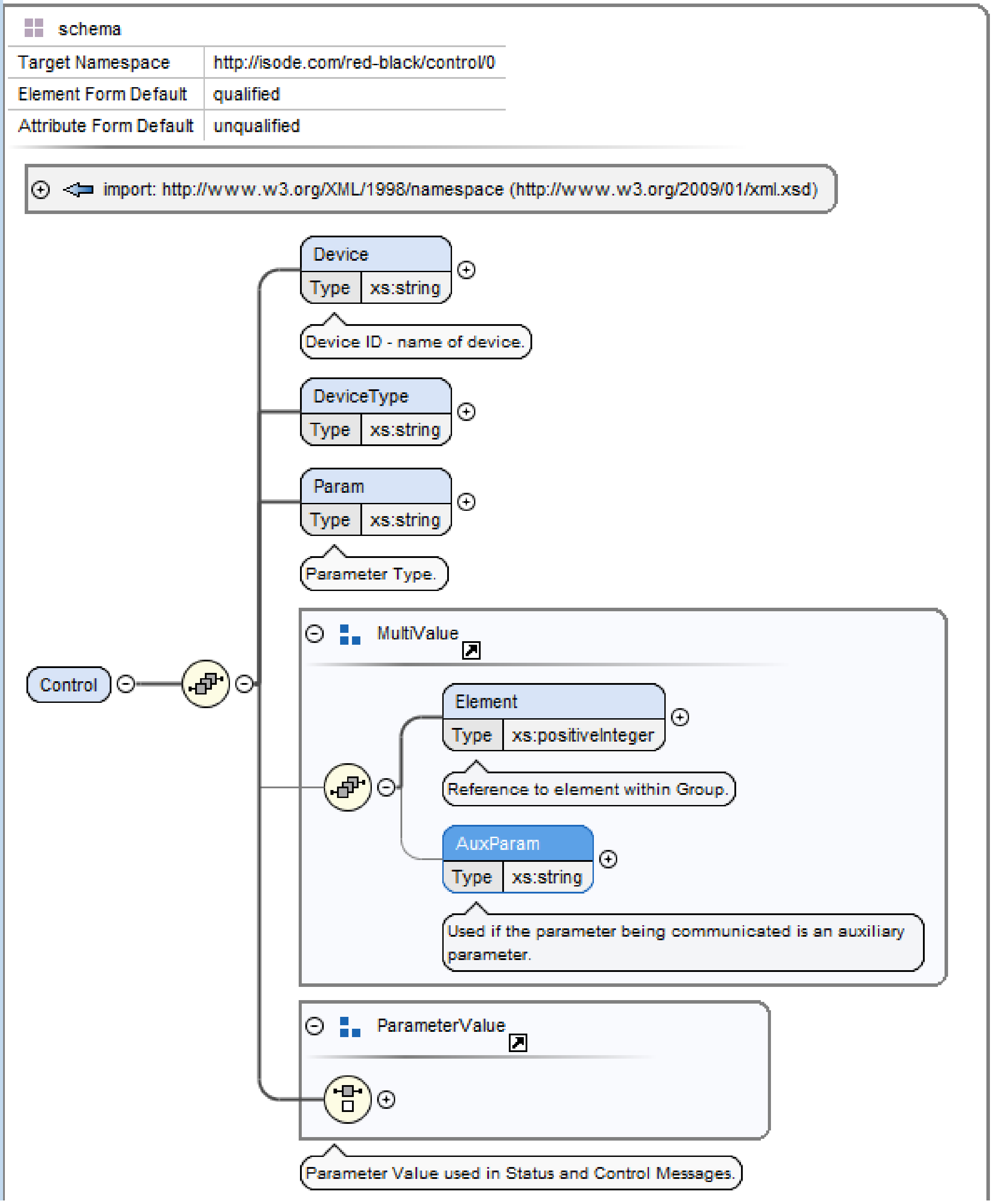
This section shows some sample messages, correctly normalized following this specification, but folded to make easier to read.

```
<Control xmlns="http://isode.com/red-black/control/0">
<Device>Modem 5</Device>
<DeviceType>Modem</DeviceType>
<Param>Speed</Param>
<Integer>1200</Integer></Control>
```

7.3.2 Schema Visualisation

This section shows the schema, as visualised by the oXygen XML editor.

This is the overall schema showing structure of the messages



schema

Target Namespace	http://isode.com/red-black/control/0
Element Form Default	qualified
Attribute Form Default	unqualified

import: http://www.w3.org/XML/1998/namespace (http://www.w3.org/2009/01/xml.xsd)

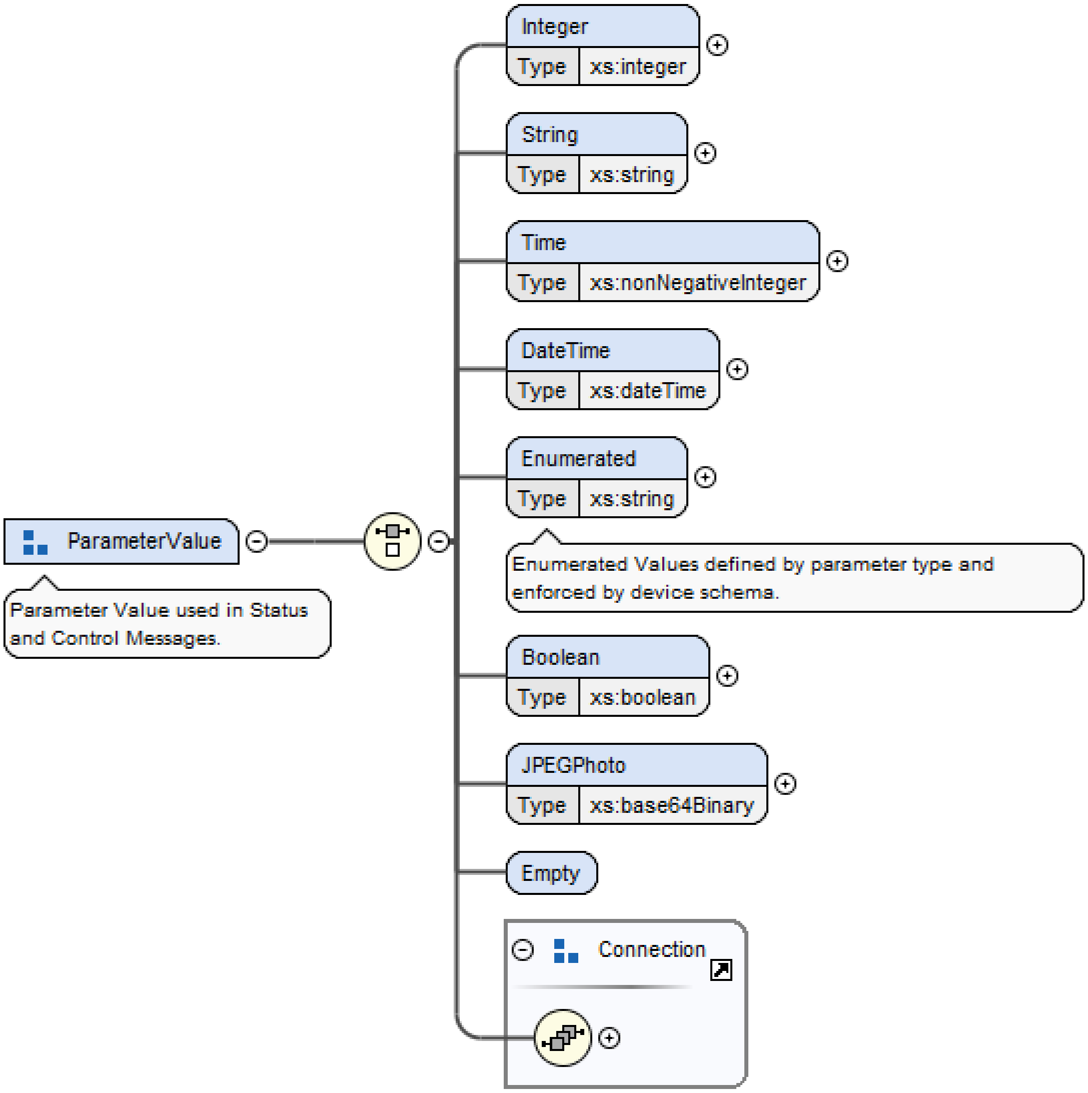
Device ID - name of device.

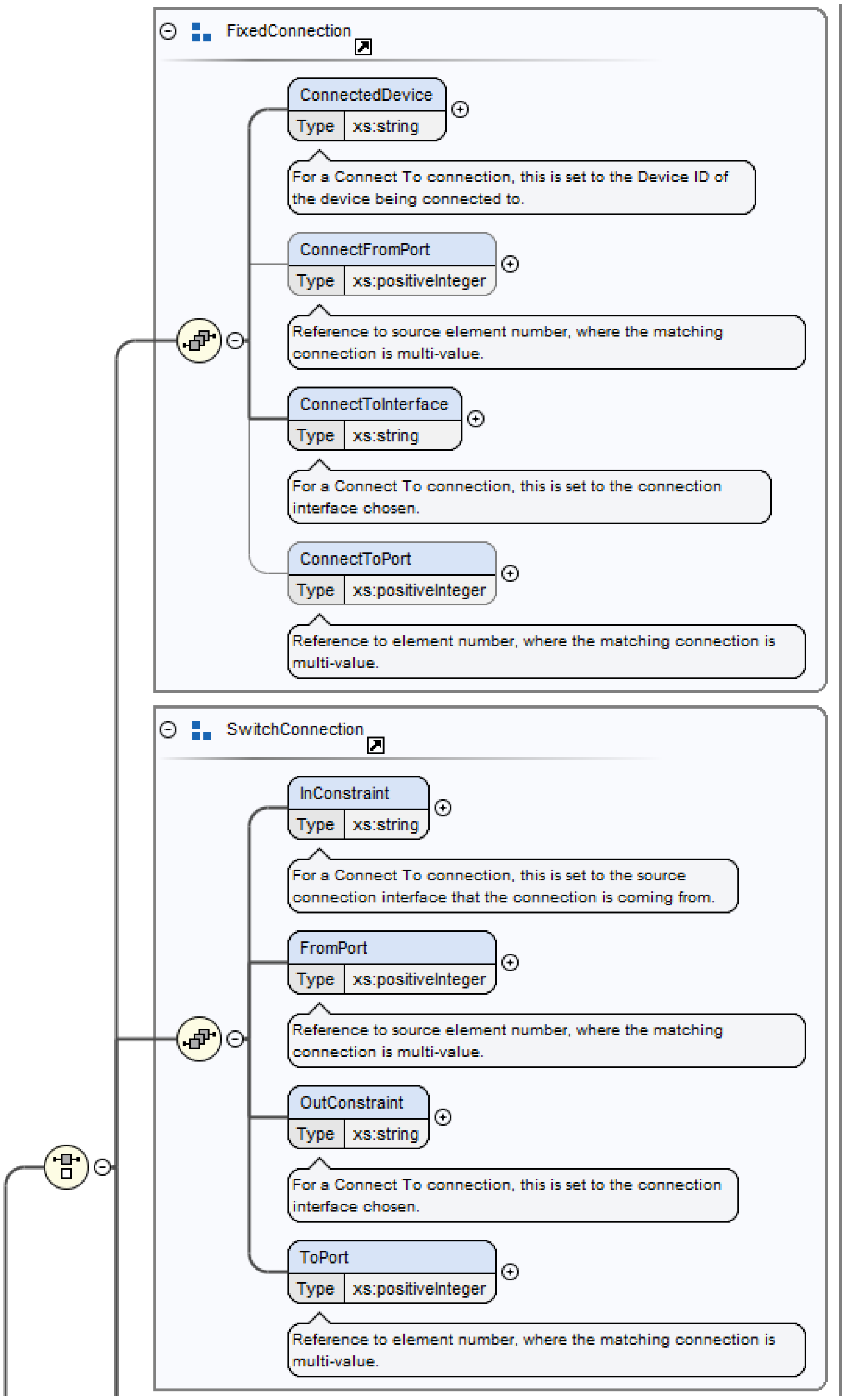
Parameter Type.

Reference to element within Group.

Used if the parameter being communicated is an auxiliary parameter.

Parameter Value used in Status and Control Messages.


Integer
Type xs:integer
String
Type xs:string
Time
Type xs:nonNegativeInteger
DateTime
Type xs:dateTime
Enumerated
Type xs:string
Enumerated Values defined by parameter type and enforced by device schema.
ParameterValue
Parameter Value used in Status and Control Messages.
Boolean
Type xs:boolean
JPEGPhoto
Type xs:base64Binary
Empty
Connection



FixedConnection

ConnectedDevice
Type xs:string

For a Connect To connection, this is set to the Device ID of the device being connected to.

ConnectFromPort
Type xs:positiveInteger

Reference to source element number, where the matching connection is multi-value.

ConnectToInterface
Type xs:string

For a Connect To connection, this is set to the connection interface chosen.

ConnectToPort
Type xs:positiveInteger

Reference to element number, where the matching connection is multi-value.

SwitchConnection

InConstraint
Type xs:string

For a Connect To connection, this is set to the source connection interface that the connection is coming from.

FromPort
Type xs:positiveInteger

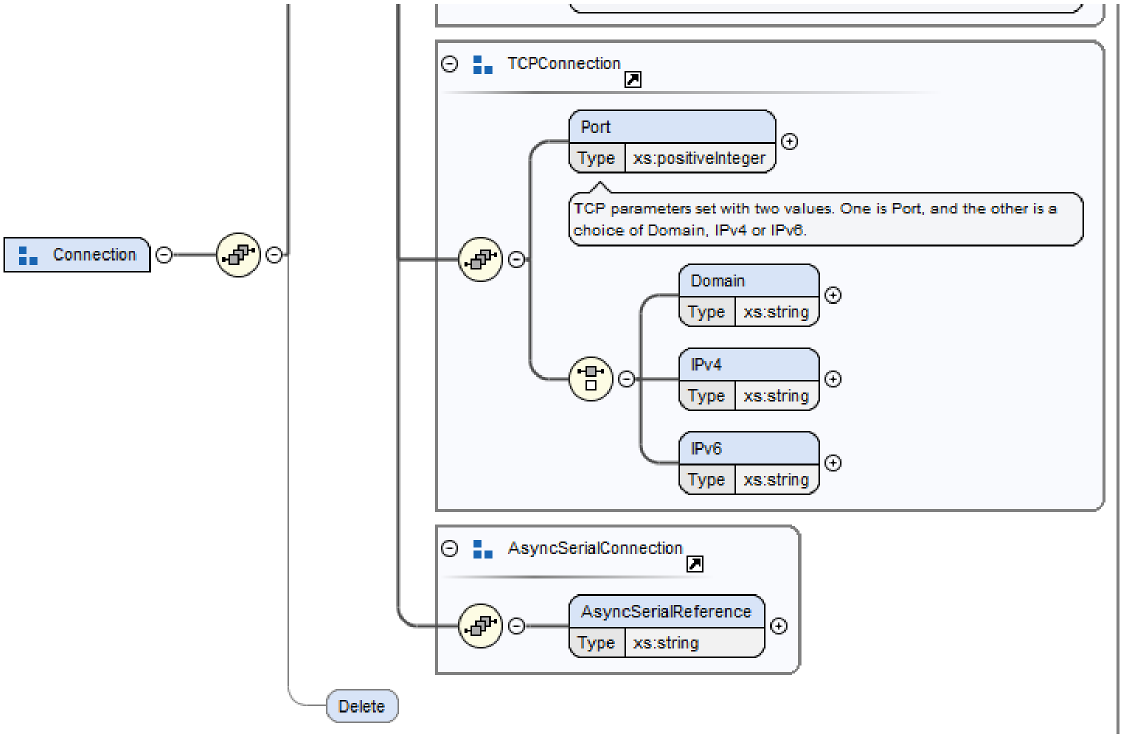
Reference to source element number, where the matching connection is multi-value.

OutConstraint
Type xs:string

For a Connect To connection, this is set to the connection interface chosen.

ToPort
Type xs:positiveInteger

Reference to element number, where the matching connection is multi-value.



7.3.3 Schema Specification

This is the formal XML Schema Definition:

```
<<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://isode.com/red-
black/control/0" xml:lang="en" targetNamespace="http://isode.com/red-black/control/0"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
    <xs:element name="Control">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Device" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Device ID - name of device.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="DeviceType" type="xs:string"/>
                <xs:element name="Param" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Parameter Type.
            </xs:documentation>
```

```
                </xs:annotation>
            </xs:element>
            <xs:group ref="MultiValue" minOccurs="0"/>
            <xs:group ref="ParameterValue"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:group name="MultiValue">
    <xs:sequence>
        <xs:element name="Element" type="xs:positiveInteger">
            <xs:annotation>
                <xs:documentation>Reference to element within Group.
      </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="AuxParam" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Used if the parameter being communicated is an
        auxiliary parameter.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:group>
<xs:group name="ParameterValue">
    <xs:annotation>
        <xs:documentation>Parameter Value used in Status and Control Messages.
  </xs:documentation>
    </xs:annotation>
    <xs:choice>
        <xs:element name="Integer" type="xs:integer"/>
        <xs:element name="String" type="xs:string"/>
        <xs:element name="Time" type="xs:nonNegativeInteger"/>
        <xs:element name="DateTime" type="xs:dateTime"/>
        <xs:element name="Enumerated" type="xs:string">
            <xs:annotation>
                <xs:documentation>Enumerated Values defined by parameter type and
         enforced by device schema.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Boolean" type="xs:boolean"/>
        <xs:element name="JPEGPhoto" type="xs:base64Binary"/>
        <xs:element name="Empty">
```

```
                <xs:complexType/>
            </xs:element>
            <xs:group ref="Connection"/>
        </xs:choice>
    </xs:group>
    <xs:group name="Connection">
        <xs:sequence>
            <xs:choice>
                <xs:group ref="FixedConnection"/>
                <xs:group ref="SwitchConnection"/>
                <xs:group ref="TCPConnection"/>
                <xs:group ref="AsyncSerialConnection"/>
            </xs:choice>
            <xs:element name="Delete" minOccurs="0">
                <xs:complexType/>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="SwitchConnection">
        <xs:sequence>
            <xs:element name="InConstraint" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the source connection interface that
the
            connection is coming from.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="FromPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to source element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="OutConstraint" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the connection interface
chosen.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ToPort" type="xs:positiveInteger">
```

```
                <xs:annotation>
                    <xs:documentation>Reference
            to element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="FixedConnection">
        <xs:sequence>
            <xs:element name="ConnectedDevice" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the Device ID of the device being
connected to. </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="ConnectFromPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to source element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ConnectToInterface" type="xs:string">
                <xs:annotation>
                    <xs:documentation>For
            a Connect To connection, this is set to the connection interface
chosen.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="ConnectToPort" type="xs:positiveInteger">
                <xs:annotation>
                    <xs:documentation>Reference
            to element number, where the matching connection is multi-
value.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:group>
    <xs:group name="TCPConnection">
        <xs:sequence>
            <xs:element name="Port" type="xs:positiveInteger">
                <xs:annotation>
```

```
                    <xs:documentation>TCP

            parameters set with two values. One is Port, and the other is a choice of
Domain, IPv4 or

            IPv6.</xs:documentation>

                </xs:annotation>

            </xs:element>

            <xs:choice>

                <xs:element name="Domain" type="xs:string"/>

                <xs:element name="IPv4" type="xs:string"/>

                <xs:element name="IPv6" type="xs:string"/>

            </xs:choice>

        </xs:sequence>

    </xs:group>

    <xs:group name="AsyncSerialConnection">

        <xs:sequence>

            <xs:element name="AsyncSerialReference" type="xs:string"/>

        </xs:sequence>

    </xs:group>

</xs:schema>
```