OTV REST API Guide

On-Ramp Wireless, Inc.
10920 Via Frontera, Suite 200
San Diego, CA 92127
U.S.A.

The information disclosed in this document is proprietary to On-Ramp Wireless, Inc. and is not to be used or disclosed to unauthorized persons without the written consent of On-Ramp Wireless, Inc. The recipient of this document shall respect the security of this document and maintain the confidentiality of the information it contains. The master copy of this document is stored in electronic format, therefore any hard or soft copy used for distribution purposes must be considered as uncontrolled. Reference should be made to On-Ramp Wireless, Inc. to obtain the latest version. By accepting this material the recipient agrees that this material and the information contained therein is to be held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of On-Ramp Wireless, Inc.

On-Ramp Wireless, Inc. reserves the right to make changes to the product(s) or information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an “as is” basis.

This document contains On-Ramp Wireless, Inc. proprietary information and must be shredded when discarded.

This documentation and the software described in it are copyrighted with all rights reserved. This documentation and the software may not be copied, except as otherwise provided in your software license or as expressly permitted in writing by On-Ramp Wireless, Inc.

Any sample code herein is provided for your convenience and has not been tested or designed to work on any particular system configuration. It is provided “AS IS” and your use of this sample code, whether as provided or with any modification, is at your own risk. On-Ramp Wireless, Inc. undertakes no liability or responsibility with respect to the sample code, and disclaims all warranties, express and implied, including without limitation warranties on merchantability, fitness for a specified purpose, and infringement. On-Ramp Wireless, Inc. reserves all rights in the sample code, and permits use of this sample code only for educational and reference purposes.

This technology and technical data may be subject to U.S. and international export, re-export or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

RPMA® (Random Phase Multiple Access) is a registered trademark of On-Ramp Wireless, Inc.

Other product and brand names may be trademarks or registered trademarks of their respective owners.

OTV REST API Guide

010-0038-00 Rev. C

November 5, 2015

Contents

Figures

Tables

Revision History

Revision	Release Date	Change Description
A	June 21, 2013	Initial release.
B	February 24, 2014	Re-organized and clarified information throughout the document.
C	October 2, 2015	Updated for OTV 1.2.

1 Introduction

The On-Ramp Wireless Total View (OTV) Representational State Transfer (REST) API provides an interface to query data from the OTV database. This API is considered “RESTful” because it adheres to the rules of data access and protocol prescribed by REST architecture. In general, a RESTful application allows machine-to-machine access to web application data via standard HTTP URLs.

2 Configuration and Security

2.1 Configuration

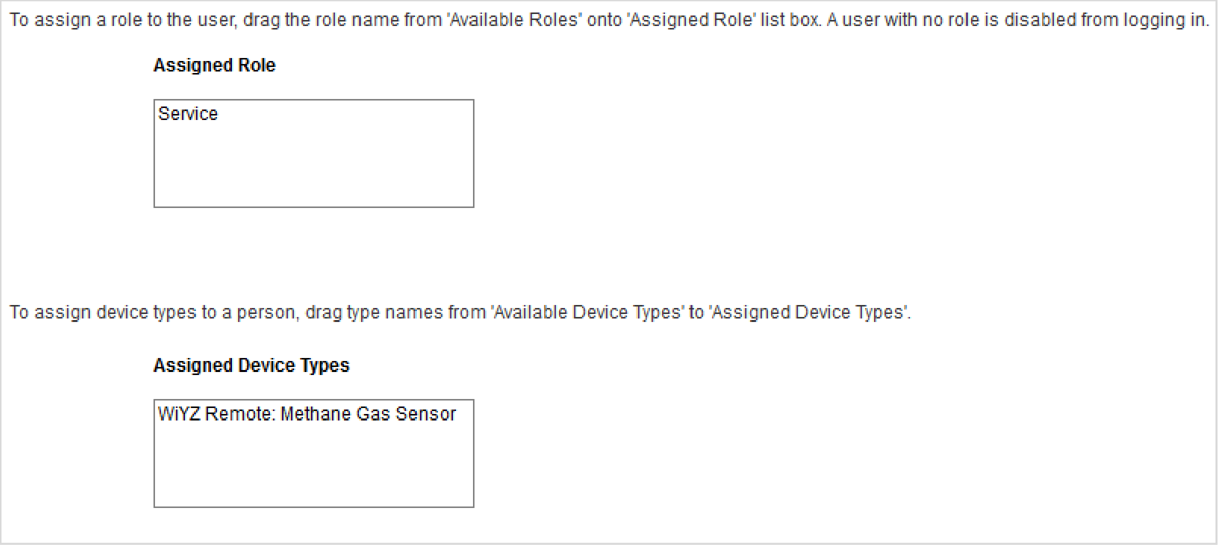
The OTV web core application must be installed to use the REST interface. Overall application properties for On-Ramp Wireless REST services are configured via the OTV config.properties file in the OTV instance directory. Editing these properties requires SSH (Secure Shell) access.

Table 1. Global Configuration Properties

Property	Description	Default
rest.services.enabled	Enables/disables overall service	true
rest.services.thread.interval	Sets the minimum seconds allowed between requests (per user)	0
rest.services.maxresults	Limits the result size for a query	50
rest.services.session.token.life	Sets the number of hours a token is valid for	24
rest.services.session.token.autoextend	This extends the life of the token each time a user uses the service	true

2.2 Setting Up a User Account

Any application that consumes On-Ramp Wireless REST services needs a valid OTV user account with a "Service" role. Users can be added/edited by an Administrator user by navigating to Admin → Users in the OTV user interface. The User must be assigned a device type in order to query the data for that particular device type.



To assign a role to the user, drag the role name from 'Available Roles' onto 'Assigned Role' list box. A user with no role is disabled from logging in.

Assigned Role

Service

To assign device types to a person, drag type names from 'Available Device Types' to 'Assigned Device Types'.

Assigned Device Types

WiYZ Remote: Methane Gas Sensor

Figure 1. Editing User for Service Role

3 Resource Overview

3.1 Root Path

The root path for all REST services is /otv/rest/. For example, http(s)://{host}:{port}/otv/rest/.

3.2 Resource Paths

All supported resource paths for the currently running OTV installation can be retrieved through the /api url. This resource returns an html file. A token is not required.

- ***Request URL***

  https://{host}:{port}/otv/rest/api

- ***Request Type***

  HTTP GET

Example

- ***Request URL***

  https://localhost:8091/otv/rest/api

3.3 Response Codes

All positive responses return an HTTP status code of 200 (OK).

3.4 Query Parameters

Parameters are optional and can be appended to the token (followed by "/?") in any order. Parameters must be separated by the "&" character. If the "apptypeid" parameter is not present, then the response only contains unparsed raw application data.

Table 2. REST Query Parameters

Parameter	Description	Value Type
datatypeid=<value>	Restricts to SDUs that have the same application/device type ID.	Integer/long
parsed=<value>	Returns parsed application data if true. Returns raw hex data if false.	Boolean
apptypeid=<value>	Deprecated. Restricts to SDUs that have the same application type ID. Returns parsed data. Equivalent to datatypeid + parsed=true.	Integer/long
id=<value>	Returns all nodes with an ID greater/equal to <value>.	Integer/long

Parameter	Description	Value Type
received=<value>	Returns all nodes that were received greater/equal than the date (epoch long).	Integer/long (milliseconds since epoch)
nodeid=<value>	Restricts to just SDUs belonging to UlpNodeId of <value>.	Integer/long
deviceid=<value>	Only for use with applications that specify a deviceid in addition to nodeid. For example, TIQ.	Integer/long
parentid=<value>	For use when retrieving downlink SDU data sent via multicast group.	Integer/long
asc=<value>	Orders results by ascending <value> where <value> is another valid parameter.	String
dsc=value>	Orders results by descending <value> where <value> is another valid parameter.	String
firstresult=<value>	Sets the starting entry in the results list to <value>. This would be used for paging purposes.	Integer/Long
maxresults=<value>	Restricts results to <value> but this does not override the OTV c onfig.properties value for maxresults.	Integer/long
name=<value>	Used for multicast group management	String

4 Establishing a Session

An OTV REST session must be established before any other resource urls are attempted. This is done by posting user credentials to the /login resource. The user name and password must be a valid OTV user with REST services privileges. After a valid login, a token is returned for all subsequent HTTP requests. The token is a hexadecimal number with 32 characters. The token is required and must be appended to the resource URL. An example is provided below.

- ***Request URL***

  https://{host}:{port}/otv/rest/login

- ***Request Type***

  HTTP POST

- ***Request Body XML Schema***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="user" type="credentials"/>
  <xs:complexType name="credentials">
    <xs:sequence>
      <xs:element name="username" type="xs:string" minOccurs="0"/>
      <xs:element name="password" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

- ***Response Body XML Schema***

```
<xs:element name="restsession" type="restToken"/>
<xs:complexType name="restToken">
    <xs:all>
      <xs:element name="expiration" type="xs:dateTime" nillable="true"/>
      <xs:element name="token" type="xs:string" nillable="true"/>
    </xs:all>
</xs:complexType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/rest/login

- ***Request POST Body***

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
    <username>rest</username>
    <password>password</password>
</user>
```

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<restsession>
    <expiration>2013-05-22T08:02:08.643-07:00</expiration>
    <token>000000003ec489b7013ec78527100010</token>
</restsession>
```

5 Obtaining Server Time

The server time is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC. This can be retrieved through the /server/time resource with a valid token from a successful login request.

- ***Request URL***

  https://{host}:{port}/otv/rest/server/time/{token}

- ***Request Type***

  HTTP GET

- ***Response Body XML Schema***

```
<xs:element name="ServerTime" type="serverTime"/>
<xs:complexType name="serverTime">
    <xs:sequence>
      <xs:element name="milliseconds" type="xs:long" nillable="true"/>
    </xs:sequence>
</xs:complexType>
```

Example

- ***Request URL***

  https://localhost:8091 /otv/rest/server/time/000000003ec489b7013ec78527100010

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServerTime>
    <milliseconds>1369153544031</milliseconds>
</ServerTime>
```

6 Retrieving Uplink Data

The /sdu/uplink/ resource is used to retrieve messages sent to the network by a device. The supported query parameters are as follows:

`datatypeid, parsed, apptypeid, id, received, nodeid, deviceid, asc, dsc, firstresult, maxresults`

See Table 2. REST Query Parameters for more details.

- ***Request URL***

  https://{host}:{port}/otv/rest/sdu/uplink/{token}

- ***Request Type***

  HTTP GET

- ***Response Body XML Schema (RawSdu)***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="RawSdu" type="rawSdu"/>

  <xs:complexType name="rawSdu">
    <xs:all>
      <xs:element name="dataTypeId" type="xs:int"/>
      <xs:element name="sduHex" type="xs:string"/>
      <xs:element name="id" type="xs:long"/>
      <xs:element name="nodeId" type="xs:long"/>
      <xs:element name="received" type="xs:dateTime"/>
      <xs:element name="ulpNodeId" type="xs:long"/>
    </xs:all>
  </xs:complexType></xs:schema>
```

Example (RawSdu – no device type specified)

- ***Request URL (no device type specified)***

  https://locahost:8091/ otv/rest/sdu/uplink/000000003ec489b7013ec78527100010?maxresults=1

- ***Response Body (no device type specified)***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xmlListWrapper>
    <size>1</size>
    <RawSdu>
        <dataTypeId>10</dataTypeId>
        <sduHex>3C3F786D6C207665</sduHex>
        <id>1</id>
        <nodeId>1</nodeId>
        <received>2013-02-22T17:33:27-08:00</received>
        <ulpNodeId>199793</ulpNodeId>
    </RawSdu>
</xmlListWrapper>
```

Example (apptypeid=10, ElectricAMISdu specified)

- ***Request URL(apptypeid=10)***

  https://localhost:8091/otv/rest/sdu/uplink/000000003ec489b7013ec78527100010?datatypeid=10&parsed=true&maxresults=1

- ***Response Body (apptypeid=10)***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xmlListWrapper>
    <size>1</size>
    <ElectricAMISdu>
        <AppTypeID>10</AppTypeID>
        <App_Device_Id>199793</App_Device_Id>
        <eventData />
        <Failures/>
        <messageSummary>Received Meter information, Serial Number:
56138508        </messageSummary>
        <Device_Id>1</Device_Id>
        <readingData />
        <Message_Received>1361583207000</Message_Received>
        <ULP_Node_Id>199793</ULP_Node_Id>
        <UplinkType>METER_INFO</UplinkType>
        <MetaAttribs/>
    </ElectricAMISdu>
</xmlListWrapper>
```

Special Note #1

If a device becomes unreachable, meaning OTV has not received an uplink message within a certain configurable time period, an empty uplink sdu entry is created in the database. Thus, when querying for uplink messages via the REST interface, you may see a RawSdu entry as shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xmlListWrapper>
      <size>1</size>
      <RawSdu>
      <dataTypeId>20</dataTypeId>
      <sduHex>00</sduHex>
      <id>1</id>
      <nodeId>1</nodeId>
      <received>2013-01-01T10:30:25-08:00</received>
      <ulpNodeId>111</ulpNodeId>
      </RawSdu>
</xmlListWrapper>
```

Special Note #2

It is recommended that the requestor of uplink messages should keep track of the last message id received for each request and make subsequent requests starting from that message id, using the “id” query parameter. This allows paged access to uplink messages that have not been retrieved yet.

7 Sending Downlink Data

Messages can be sent to devices using the /sdu/downlink/ resource. The supported query parameters are: `nodeid, deviceid.`

- ***Request URL***

  https://{host}:{port}/otv/rest/sdu/downlink/{token}

- ***Request Type***

  HTTP POST

- ***Request Body XML Schema***

```
<xs:element name="downlinkrequest">
    <xs:complexType>
      <xs:all>
        <xs:element name="description" type="xs:string" nillable="true"/>
        <xs:element name="flush" type="xs:boolean" nillable="true"/>
        <xs:element name="payload" type="xs:string" nillable="true"/>
        <xs:element name="UlpNodeId" type="xs:long" nillable="true"/>
        <xs:element name="ping" type="xs:boolean" nillable="true"/>
        <xs:element name="streamId" type="xs:int" nillable="true"/>
      </xs:all>
    </xs:complexType>
</xs:element>
```

The payload must be in hexadecimal characters and may include spaces; all white space are deleted before the message is sent.

Example (hex payload)

- ***Request URL***

  https://localhost:8091/otv/rest/sdu/downlink/000000003ec489b7013ec78527100010

- ***Request POST Body***

```
<?xml version="1.0" encoding="UTF-8"?>
<downlinkrequest>
      <UlpNodeId>111</UlpNodeId>
      <payload>24 FF 01 70 C0 03 01 06</payload>
      <flush>false</flush>
</downlinkrequest>
```

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<DownlinkSdu>
      <App_Device_Id>111</App_Device_Id>
      <dataTypeId>111</dataTypeId>
      <payload>
       <downlinkMessageParams>
            <hex>24FF0170C0030106</hex>
            <name>dl_payload</name>
       </downlinkMessageParams>
      </payload>
<Message_Id>11</Message_Id>
```

```
<Device_Id>6</Device_Id>
<Message_Received>1389048679000</Message_Received>
<ackstatus>-1</ackstatus>
<ULP_Node_Id>111</ULP_Node_Id>
</DownlinkSdu>
```

Example (flush)

- ***Request URL***

  https://localhost:8091/otv/rest/sdu/downlink/000000003ec489b7013ec78527100010

- ***Request POST Body***

```
<?xml version="1.0" encoding="UTF-8"?>
<downlinkrequest>
      <UlpNodeId>111</UlpNodeId>
      <flush>true</flush>
</downlinkrequest>
```

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<DownlinkSdu>
      <App_Device_Id>111</App_Device_Id>
      <dataTypeId>111</dataTypeId>
      <payload>
       <downlinkMessageParams>
            <hex/>
            <name>flush</name>
       </downlinkMessageParams>
      </payload>
<Message_Id>11</Message_Id>
<Device_Id>6</Device_Id>
<Message_Received>1389048679000</Message_Received>
<ackstatus>-1</ackstatus>
<ULP_Node_Id>111</ULP_Node_Id>
</DownlinkSdu>
```

Example (ping)

- ***Request URL***

  https://localhost:8091/otv/rest/sdu/downlink/000000003ec489b7013ec78527100010

- ***Request POST Body***

```
<?xml version="1.0" encoding="UTF-8"?>
<downlinkrequest>
      <UlpNodeId>111</UlpNodeId>
      <ping>true</flush>
</downlinkrequest>
```

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<DownlinkSdu>
      <App_Device_Id>111</App_Device_Id>
      <dataTypeId>111</dataTypeId>
      <payload>
       <downlinkMessageParams>
```

```
            <hex/>
            <name>ping</name>
       </downlinkMessageParams>
      </payload>
<Message_Id>11</Message_Id>
<Device_Id>6</Device_Id>
<Message_Received>1389048679000</Message_Received>
<ackstatus>-1</ackstatus>
<ULP_Node_Id>111</ULP_Node_Id>
</DownlinkSdu>
```

NOTE: The status of downlink messages sent can be retrieved by making a GET request to the /sdu/downlink resource using the message id returned in this response. See Chapter 8: Retrieving Downlink Data for more details.

8 Retrieving Downlink Data

The /sdu/downlink resource is also used to retrieve messages that have already been sent to a device from the network.

- ***Request URL***

  https://{host}:{port}/otv/rest/sdu/downlink/{token}

  The supported query parameters are as follows:

  datatypeid, apptypeid, id, received, nodeid, deviceid, parentid, asc, dsc, firstresult, maxresults

  See Table 2. REST Query Parameters for more details.

- ***Request Type***

  HTTP GET

- ***Response Body XML Schema***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="DownlinkSdu" type="downlinkSdu"/>

  <xs:complexType name="downlinkSdu">
    <xs:all>
      <xs:element name="App_Device_Id" type="xs:long"/>
      <xs:element name="dataTypeId" type="xs:int"/>
      <xs:element name="payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="downlinkMessageParams"
type="downlinkSduRawChunk" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Message_Id" type="xs:long"/>
      <xs:element name="Device_Id" type="xs:long"/>
      <xs:element name="parentId" type="xs:long"/>
      <xs:element name="Message_Received" type="xs:long"/>
      <xs:element name="ackstatus" type="xs:string"/>
      <xs:element name="ULP_Node_Id" type="xs:long"/>
      <xs:element name="streamId" type="xs:int"/>
      </xs:all>
  </xs:complexType>

  <xs:complexType name="downlinkSduRawChunk">
    <xs:sequence>
      <xs:element name="hex" type="xs:string" minOccurs="0"/>
      <xs:element name="name" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ackStatusCodes">
    <xs:restriction base="xs:string">
      <xs:enumeration value="-1 PENDING"/>
      <xs:enumeration value="0 ACKNOWLEDGED"/>
      <xs:enumeration value="1 UNKNOWN"/>
      <xs:enumeration value="2 DROPPED"/>
      <xs:enumeration value="3 NODE_NOT_CONNECTED"/>
      <xs:enumeration value="4 UNKNOWN_NODE_ID"/>
      <xs:enumeration value="5 INVALID_PAYLOAD_SIZE"/>
      <xs:enumeration value="6 QUEUE_FULL"/>
      <xs:enumeration value="7 UNKNOWN_GROUP_ID"/>
      <xs:enumeration value="8 SENT"/>
      <xs:enumeration value="9 INVALID MESSAGE"/>
      <xs:enumeration value="10 SUCCESS"/>
      <xs:enumeration value="11 TAG_ALREADY_EXISTS"/>
      <xs:enumeration value="12 DL_DROPPED"/>
      <xs:enumeration value="20 REACHABLE"/>
      <xs:enumeration value="21 NOT_REACHABLE"/>
      <xs:enumeration value="22 ECHO_NOT_SUPPORTED"/>
      <xs:enumeration value="128 APP_TIMEOUT"/>
      <xs:enumeration value="129 APP_GW_UNREACHABLE"/>
      <xs:enumeration value="130 APP_PROCESS_ERROR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Example

- ***Request URL***

  https://localhost:8091/otv/rest/sdu/downlink/000000003ec489b7013ec78527100010?nodeid=510&maxresults=1

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xmlListWrapper xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <size>1</size>
    <DownlinkSdu>
        <App_Device_Id>111</App_Device_Id>
        <dataTypeId>111</dataTypeId>
        <payload>
            <downlinkMessageParams>
                <hex>DEADBEEF</hex>
                <name>dl_payload</name>
            </downlinkMessageParams>
        </payload>
        <Message_Id>8</Message_Id>
        <Device_Id>6</Device_Id>
        <Message_Received>1388776239000</Message_Received>
        <ackstatus>-1</ackstatus>
        <ULP_Node_Id>111</ULP_Node_Id>
    </DownlinkSdu>
</xmlListWrapper>
```

9 Retrieving Devices

A list of devices associated to a node id may be retrieved using the following request:

- ***Request URL***

  https://{host}:{port}/otv/rest/device/{token}?nodeid={nodeid}

- ***Request Type***

  HTTP GET

- ***Response Body XML Schema***

```
<xsi:element name="devicelist" type="deviceList"/>

<xsi:complexType name="deviceList">
    <xsi:sequence>
      <xsi:element ref="device" minOccurs="0" maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="device">
    <xsi:sequence>
      <xsi:element name="deviceid" type="xsi:long"/>
      <xsi:element name="appTypeId" type="xsi:int"/>
      <xsi:element name="nodeid" type="xsi:long"/>
    </xsi:sequence>
  </xsi:complexType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/rest/device/000000003ec489b7013ec78527100010?nodeid=8604

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<devicelist xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <device>
        <deviceid>199205</deviceid>
        <appTypeId>30</appTypeId>
        <nodeid>199205</nodeid>
    </device>
</devicelist>
```

10 Retrieving Device Metadata

Devices may have metadata, in the form of key value pairs, stored for searching or filtering or other application specific information. The /meta/ resource can be used to retrieve existing metadata stored for a device.

- ***Request URL (using node id identifier)***

  https://{host}:{port}/otv/rest/meta/{token}?nodeid={nodeid}

- ***Request URL (using other device identifier, such as meter serial number)***

  https://{host}:{port}/otv/rest/device/{key}/{value}/meta/{token}

- ***Request Type***

  HTTP GET

- ***Response Body XML Schema***

```
<xs:element name="MetaData" type="metaData"/>

<xs:complexType name="Attribute">
    <xs:sequence>
      <xs:element name="Name" type="xs:string" nillable="true"/>
      <xs:element name="Value" type="xs:string" nillable="true"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="metaData">
    <xs:sequence>
      <xs:element name="Attributes" type="Attribute" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

Example

- ***Request URL (using node id)***

  https://localhost:8091/otv/rest/meta/000000003ec489b7013ec78527100010?nodeid=8604

- ***Request URL (using HES/MDMS Meter ID)***

  https://localhost:8091/otv/rest/device/HES%2FMDMS%20Meter%20ID/54534672/meta/000000003ec489b7013ec78527100010

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<MetaData>
 <Attribute>
      <Name>Description</Name>
      <Value>this is the description</Value>
 </Attribute>
 <Attribute>
```

```
      <Name>Latitude</Name>
      <Value />
 </Attribute>
 <Attribute>
      <Name>Longitude</Name>
      <Value />
 </Attribute>
</MetaData>
```

11 Updating Device Metadata

Device metadata may be updated by submitting a PUT request on the /meta resource. This does not do a complete replacement of all metadata but only updates/adds the attributes provided in the PUT XML body.

- ***Request URL (using node id identifier)***

  https://{host}:{port}/otv/rest/meta/{token}?nodeid={nodeid}

- ***Request URL (using other device identifier, such as meter serial number)***

  https://{host}:{port}/otv/rest/device/{key}/{value}/meta/{token}

- ***Request Type***

  HTTP PUT

- ***Request and Response Body XML Schema***

```
<xs:element name="MetaData" type="metaData"/>

<xs:complexType name="Attribute">
    <xs:sequence>
      <xs:element name="Name" type="xs:string" nillable="true"/>
      <xs:element name="Value" type="xs:string" nillable="true"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="metaData">
    <xs:sequence>
      <xs:element name="Attributes" type="Attribute" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

Example

- ***Request URL (using node id)***

  https://localhost:8091/otv/rest/meta/000000003ec489b7013ec78527100010?nodeid=8604

- ***Request URL (using HES/MDMS Meter ID)***

  https://localhost:8091/otv/rest/device/HES%2FMDMS%20Meter%20ID/54534672/meta/000000003ec489b7013ec78527100010

- ***Request PUT Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MetaData xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Attribute>
        <Name>Description</Name>
        <Value>this is the updated description</Value>
    </Attribute>
</MetaData>
```

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MetaData>
    <Attribute>
        <Name>Description</Name>
        <Value>this is the updated description</Value>
    </Attribute>
    <Attribute>
        <Name>Latitude</Name>
        <Value></Value>
    </Attribute>
    <Attribute>
        <Name>Longitude</Name>
        <Value></Value>
    </Attribute>
</MetaData>
```

12 Deleting Device Metadata

Device metadata may be updated by submitting a DELETE request on the /meta resource. This deletes all metadata associated with the device.

- ***Request URL (using node id identifier)***

  https://{host}:{port}/otv/rest/meta/{token}?nodeid={nodeid}

- ***Request URL (using other device identifier, such as meter serial number)***

  https://{host}:{port}/otv/rest/device/{key}/{value}/meta/{token}

- ***Request Type***

  HTTP DELETE

- ***Response Body XML Schema***

```
<xs:element name="MetaData" type="metaData"/>

<xs:complexType name="Attribute">
    <xs:sequence>
      <xs:element name="Name" type="xs:string" nillable="true"/>
      <xs:element name="Value" type="xs:string" nillable="true"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="metaData">
    <xs:sequence>
      <xs:element name="Attributes" type="Attribute" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

Example

- ***Request URL (using node id)***

  https://localhost:8091/otv/rest/meta/000000003ec489b7013ec78527100010?nodeid=8604

- ***Request URL (using HES/MDMS Meter ID)***

  https://localhost:8091/otv/rest/device/HES%2FMDMS%20Meter%20ID/54534672/meta/000000003ec489b7013ec78527100010

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<MetaData />
```

13 Multicast Group Creation

Multicast groups must be created on the Gateway. This REST resource sends requests to the On-Ramp Gateway service. Groups must be created before devices are assigned to the groups.

- ***Request URL***

  https://{host}:{port}/otv/rest/group/create/{token}?name={name}

- ***Request Type***

  HTTP POST

- ***Response Body XML Schema***

```
<xsi:complexType name="genericRestResponse">
    <xsi:sequence>
      <xsi:element name="message" type="xsi:string" minOccurs="0"/>
      <xsi:element name="result" type="code" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="code">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PROCESSED"/>
      <xsi:enumeration value="DROPPED"/>
    </xsi:restriction>
  </xsi:simpleType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/group/create/000000003ec489b7013ec78527100010?name=test_mud_group

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
   <Response>
     <message>
     Request sent for processing, created groupid=fb32514a-3119-4fad-9d8d-
     de9ef4203bb5, for name=test_mud_group
     </message>
     <result>
     PROCESSED
     </result>
   </Response>
```

The status of this group creation request can be retrieved through the /group/membership/ resource. See Chapter 17: Multicast Group Membership for details.

14 Multicast Group Removal

Multicast groups can be removed using the following request.

- ***Request URL***

  https://{host}:{port}/otv/rest/group/destroy/{token}?name={name}

- ***Request Type***

  HTTP DELETE

- ***Response Body XML Schema***

```
<xsi:complexType name="genericRestResponse">
    <xsi:sequence>
      <xsi:element name="message" type="xsi:string" minOccurs="0"/>
      <xsi:element name="result" type="code" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="code">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PROCESSED"/>
      <xsi:enumeration value="DROPPED"/>
    </xsi:restriction>
  </xsi:simpleType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/group/destroy/000000003ec489b7013ec78527100010?name=test_mud_group

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Response xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <message>Request sent for processing for groupid=fb32514a-3119-4fad-9d8d-
de9ef4203bb5, name=test_mud_group</message>

    <result>PROCESSED</result>

</Response>
```

15 Multicast Group Assignment

Devices can be added to multicast groups using the following request.

- ***Request URL***

  https://{host}:{port}/otv/rest/group/assign/{token}?name={name}&nodeid={nodeid}

- ***Request Type***

  HTTP POST

- ***Response Body XML Schema***

```
<xsi:complexType name="genericRestResponse">
    <xsi:sequence>
      <xsi:element name="message" type="xsi:string" minOccurs="0"/>
      <xsi:element name="result" type="code" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="code">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PROCESSED"/>
      <xsi:enumeration value="DROPPED"/>
    </xsi:restriction>
  </xsi:simpleType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/group/assign/000000003ec489b7013ec78527100010?name=test_mud_group&nodeid=199205

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Response xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <message>Request sent for processing for groupid=d75ea327-8f4c-
4d00-aed4-59ed142395ae, name=test_mud_group, nodeid=199205</message>

    <result>PROCESSED</result>

</Response>
```

The status of this group assignment request can be retrieved through the /group/membership/ resource. See Chapter 17: Multicast Group Membership for details.

16 Multicast Group Deassignment

Devices can be removed from multicast groups using the following request.

- ***Request URL***

  https://{host}:{port}/otv/rest/group/unassign/{token}?name={name}&nodeid={nodeid}

- ***Request Type***

  HTTP DELETE

- ***Response Body XML Schema***

```
<xsi:complexType name="genericRestResponse">
    <xsi:sequence>
      <xsi:element name="message" type="xsi:string" minOccurs="0"/>
      <xsi:element name="result" type="code" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="code">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PROCESSED"/>
      <xsi:enumeration value="DROPPED"/>
    </xsi:restriction>
  </xsi:simpleType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/group/unassign/000000003ec489b7013ec78527100010?name=test_mud_group&nodeid=199205

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Response xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <message>Request sent for processing for groupid=d75ea327-8f4c-
4d00-aed4-59ed142395ae, name=test_mud_group, nodeid=199205</message>

    <result>PROCESSED</result>

</Response>
```

The status of this group deassignment request can be retrieved through the /group/membership/ resource. See Chapter 17: Multicast Group Membership for details.

17 Multicast Group Membership

17.1 Membership by Group

To obtain a listing of multicast groups, and the status of each device in a group, the following request may be used. If no name is specified, all groups are returned.

- ***Request URL***

  https://{host}:{port}/otv/rest/group/membership/{token}?name={name}

- ***Request Type***

  HTTP GET

- ***Response Body XML Schema***

```
  <xsi:element name="GroupMembershipList" type="mudGroupList"/>

  <xsi:element name="GroupMembership" type="mudGroupMembershipStatus"/>

  <xsi:complexType name="mudGroupList">
    <xsi:sequence>
      <xsi:element ref="GroupMembership" minOccurs="0" maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="mudGroupMembershipStatus">
    <xsi:sequence>
      <xsi:element name="name" type="xsi:string" minOccurs="0"/>
      <xsi:element name="groupid" type="xsi:string" minOccurs="0"/>
      <xsi:element name="GroupMember" type="GroupMember" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="GroupMember">
    <xsi:sequence>
      <xsi:element name="nodeid" type="xsi:long" nillable="true"/>
      <xsi:element name="status" type="mudMembershipState" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="mudMembershipState">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PENDING_ASSIGNMENT"/>
      <xsi:enumeration value="ASSIGNED"/>
      <xsi:enumeration value="PENDING_DEASSIGNMENT"/>
      <xsi:enumeration value="DEASSIGNED"/>
    </xsi:restriction>
  </xsi:simpleType>
```

Example (single group)

- ***Request URL***

  https://localhost:8091/otv/group/membership/000000003ec489b7013ec78527100010?name=test_mud_group

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GroupMembership xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <name>test_mud_group</name>
    <groupid>d75ea327-8f4c-4d00-aed4-59ed142395ae</groupid>
    <GroupMember>
        <nodeid>199205</nodeid>
        <status>ASSIGNED</status>
    </GroupMember>
</GroupMembership>
```

Example (all groups)

- ***Request URL***

https://localhost:8091/otv/group/membership/000000003ec489b7013ec78527100010

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GroupMembershipList>
<GroupMembership xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <name>test_mud_group</name>
    <groupid>d75ea327-8f4c-4d00-aed4-59ed142395ae</groupid>
    <GroupMember>
        <nodeid>199205</nodeid>
        <status>ASSIGNED</status>
    </GroupMember>
</GroupMembership>
</GroupMembershipList>
```

17.2 Membership by Node

To obtain a listing of multicast groups that a node belong to, and the status of the node in each group, the following request may be used.

- ***Request URL***

https://{host}:{port}/otv/rest/group/membership/{token}?nodeid={nodeid}

- ***Request Type***

HTTP GET

- ***Response Body XML Schema***

```
  <xsi:element name="NodeMembership" type="mudNodeMembershipStatus"/>

  <xsi:complexType name="mudNodeMembershipStatus">
    <xsi:sequence>
      <xsi:element name="id" type="xsi:long" minOccurs="0"/>
      <xsi:element name="NodeMember" type="NodeMember" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>
  <xsi:complexType name="NodeMember">
    <xsi:sequence>
      <xsi:element name="name" type="xsi:string" nillable="true"/>
      <xsi:element name="groupid" type="xsi:string" nillable="true"/>
      <xsi:element name="status" type="mudMembershipState"
nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="mudMembershipState">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PENDING_ASSIGNMENT"/>
      <xsi:enumeration value="ASSIGNED"/>
      <xsi:enumeration value="PENDING_DEASSIGNMENT"/>
      <xsi:enumeration value="DEASSIGNED"/>
    </xsi:restriction>
  </xsi:simpleType>
```

Example

- ***Request URL***

  https://localhost:8091/otv/group/membership/000000003ec489b7013ec78527100010?nodeid=199205

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NodeMembership xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>199205</id>
    <NodeMember>
        <name>test_mud_group</name>
        <groupid>d75ea327-8f4c-4d00-aed4-59ed142395ae</groupid>
        <status>ASSIGNED</status>
    </NodeMember>
</NodeMembership>
```

18 Multicast Downlink Data

Messages can be sent via multicast to devices using the /group/downlink/ resource. The supported query parameter is: `name`.

- ***Request URL***

  https://{host}:{port}/otv/rest/group/downlink/{token}?name={name}

- ***Request Type***

  HTTP POST

- ***Request Body XML Schema***

```
<xs:element name="downlinkrequest">
    <xs:complexType>
      <xs:all>
        <xs:element name="description" type="xs:string" nillable="true"/>
        <xs:element name="flush" type="xs:boolean" nillable="true"/>
        <xs:element name="payload" type="xs:string" nillable="true"/>
        <xs:element name="UlpNodeId" type="xs:long" nillable="true"/>
       <xs:element name="ping" type="xs:boolean" nillable="true"/>
       <xs:element name="streamId" type="xs:int" nillable="true"/>
      </xs:all>
    </xs:complexType>
</xs:element>
```

  The payload must be in hexadecimal characters and may include spaces; all white space are deleted before the message is sent.

Example

- ***Request URL***

  https://localhost:8091/otv/rest/sdu/downlink/000000003ec489b7013ec78527100010?name=test_mud_group

- ***Request POST Body***

```
<?xml version="1.0" encoding="UTF-8"?>
<downlinkrequest>
   <payload>24 FF 01 70 C0 03 01 06</payload>
</downlinkrequest>
```

- ***Response Body***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<DownlinkSdu>
     <App_Device_Id>0</App_Device_Id>
     <dataTypeId>111</dataTypeId>
     <payload>
        <downlinkMessageParams>
             <hex>24FF0170C0030106</hex>
             <name>dl_payload</name>
        </downlinkMessageParams>
     </payload>
<Message_Id>11</Message_Id>
```

```
<Device_Id>0</Device_Id>
<Message_Received>1389048679000</Message_Received>
<ackstatus>-1</ackstatus>
<ULP_Node_Id>0</ULP_Node_Id>
        </DownlinkSdu>
```

The status of this downlink message can be polled using the /sdu/downlink resource described in the Retrieving Downlink Data chapter.

19 DNP3 Concentrator Management

DNP3 concentrator management is documented in the DNP3 Concentrator Configuration Guide (010-0117-00).

20 Retrieving XML Schema Definitions

The /api/schema/ resource returns the xml schemas used for the REST interface. A token is not required. The query parameter “apptypeid” can be used to retrieve the schema for a specific device type. See Table 2. REST Query Parameters.

- ***Request URL***

  https://{host}:{port}/otv/rest/api/schema/

- ***Request Type***

  HTTP GET

Example

- ***Request URL (no device type specified)***

  https://localhost:8091/otv/rest/api/schema/

- ***Response Body (no device type specified)***

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="DownlinkSdu" type="downlinkSdu"/>

  <xsi:element name="GroupMembership" type="mudGroupMembershipStatus"/>

  <xsi:element name="GroupMembershipList" type="mudGroupList"/>

  <xsi:element name="MetaData" type="metaData"/>

  <xsi:element name="NodeMembership" type="mudNodeMembershipStatus"/>

  <xsi:element name="RawSdu" type="rawSdu"/>

  <xsi:element name="Response" type="genericRestResponse"/>

  <xsi:element name="ServerTime" type="serverTime"/>

  <xsi:element name="device" type="device"/>

  <xsi:element name="devicelist" type="deviceList"/>

  <xsi:element name="downlinkrequest" type="restDownlinkRequest"/>

  <xsi:element name="restsession" type="restToken"/>

  <xsi:element name="user" type="credentials"/>

  <xsi:element name="xmlListWrapper" type="xmlListWrapper"/>

  <xsi:complexType name="credentials">
    <xsi:sequence>
      <xsi:element name="username" type="xsi:string" minOccurs="0"/>
      <xsi:element name="password" type="xsi:string" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="restToken">
    <xsi:all>
      <xsi:element name="expiration" type="xsi:dateTime" nillable="true"/>
      <xsi:element name="token" type="xsi:string" nillable="true"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="serverTime">
    <xsi:sequence>
      <xsi:element name="milliseconds" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="xmlListWrapper">
    <xsi:sequence>
      <xsi:element name="size" type="xsi:int"/>
      <xsi:any processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="rawSdu">
    <xsi:all>
      <xsi:element name="dataTypeId" type="xsi:int"/>
      <xsi:element name="sduHex" type="xsi:string"/>
      <xsi:element name="id" type="xsi:long"/>
      <xsi:element name="nodeId" type="xsi:long"/>
      <xsi:element name="received" type="xsi:dateTime"/>
      <xsi:element name="ulpNodeId" type="xsi:long"/>
      <xsi:element name="ustreamId" type="xsi:int" minOccurs="0"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="downlinkSdu">
    <xsi:all>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="dataTypeId" type="xsi:int"/>
      <xsi:element name="payload">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="downlinkMessageParams" type="downlinkSduRawChunk"
nillable="true" minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="parentId" type="xsi:long" minOccurs="0"/>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="ackstatus" type="xsi:string"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="streamId" type="xsi:int" minOccurs="0"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="downlinkSduRawChunk">
    <xsi:sequence>
      <xsi:element name="hex" type="xsi:string" minOccurs="0"/>
      <xsi:element name="name" type="xsi:string" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>
  <xsi:complexType name="restDownlinkRequest">
    <xsi:all>
      <xsi:element name="description" type="xsi:string" nillable="true"
minOccurs="0"/>
      <xsi:element name="flush" type="xsi:boolean" nillable="true"/>
      <xsi:element name="UlpNodeId" type="xsi:long" nillable="true" minOccurs="0"/>
      <xsi:element name="payload" type="xsi:string" nillable="true"/>
      <xsi:element name="ping" type="xsi:boolean" nillable="true"/>
      <xsi:element name="streamId" type="xsi:int" nillable="true" minOccurs="0"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="genericRestResponse">
    <xsi:sequence>
      <xsi:element name="message" type="xsi:string" minOccurs="0"/>
      <xsi:element name="result" type="code" minOccurs="0"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="metaData">
    <xsi:sequence>
      <xsi:element name="Attribute" type="MetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="MetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="mudGroupList">
    <xsi:sequence>
      <xsi:element ref="GroupMembership" minOccurs="0" maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="mudGroupMembershipStatus">
    <xsi:sequence>
      <xsi:element name="name" type="xsi:string" minOccurs="0"/>
      <xsi:element name="groupid" type="xsi:string" minOccurs="0"/>
      <xsi:element name="GroupMember" type="GroupMember" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="GroupMember">
    <xsi:sequence>
      <xsi:element name="nodeid" type="xsi:long" nillable="true"/>
      <xsi:element name="status" type="mudMembershipState" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="mudNodeMembershipStatus">
    <xsi:sequence>
      <xsi:element name="id" type="xsi:long" minOccurs="0"/>
      <xsi:element name="NodeMember" type="NodeMember" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="NodeMember">
    <xsi:sequence>
      <xsi:element name="name" type="xsi:string" nillable="true"/>
      <xsi:element name="groupid" type="xsi:string" nillable="true"/>
      <xsi:element name="status" type="mudMembershipState" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="deviceList">
    <xsi:sequence>
      <xsi:element ref="device" minOccurs="0" maxOccurs="unbounded"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="device">
    <xsi:sequence>
      <xsi:element name="deviceid" type="xsi:long"/>
      <xsi:element name="appTypeId" type="xsi:int"/>
      <xsi:element name="nodeid" type="xsi:long"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="code">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PROCESSED"/>
      <xsi:enumeration value="DROPPED"/>
    </xsi:restriction>
  </xsi:simpleType>

  <xsi:simpleType name="mudMembershipState">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PENDING_ASSIGNMENT"/>
      <xsi:enumeration value="ASSIGNED"/>
      <xsi:enumeration value="PENDING_DEASSIGNMENT"/>
      <xsi:enumeration value="DEASSIGNED"/>
    </xsi:restriction>
  </xsi:simpleType>
</xsi:schema>
```

Appendix A Uplink SDU Schema Reference

The XML *payload* is the body data or contents (e.g., the “cargo”) of a data transmission as opposed to the *wrapper* which is the boilerplate XML around the content. As such, the payload is the *variable* part of the reply. This chapter provides the XML schemas (xsd) for uplink SDUs for various device types. The schemas can also be obtained by using the /api/schema/ resource.

A.1 WiYZSdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="WiYZProcessControlValue"
type="wiYZProcessControlValue"/>

  <xsi:element name="WiYZSdu" type="wiYZSdu"/>

  <xsi:complexType name="wiYZSdu">
    <xsi:all>
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="isAlarm" type="xsi:boolean"/>
      <xsi:element name="alertClass" type="xsi:int"/>
      <xsi:element name="alertCounter" type="xsi:int"/>
      <xsi:element name="alertDirection" type="xsi:int"/>
      <xsi:element name="alertId" type="xsi:int"/>
      <xsi:element name="alertReserved" type="xsi:int"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="sensorData">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="sensor" type="wiYZProcessControlValue"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="destinationOID" type="xsi:int"/>
      <xsi:element name="downlinkResponse" type="xsi:string"/>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="Failures">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="messageType" type="xsi:int"/>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="sduTimestamp" type="xsi:long"/>
      <xsi:element name="sourceOID" type="xsi:int"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="wiYZProcessControlValue">
    <xsi:all>
      <xsi:element name="id" type="xsi:long"/>
      <xsi:element name="limitstatus" type="xsi:int"/>
      <xsi:element name="nodeId" type="xsi:long"/>
      <xsi:element name="portId" type="xsi:long"/>
      <xsi:element name="quality" type="xsi:int"/>
      <xsi:element name="reserved" type="xsi:int"/>
      <xsi:element name="scaledValue" type="xsi:string"/>
      <xsi:element name="sduId" type="xsi:long"/>
      <xsi:element name="substatus" type="xsi:int"/>
      <xsi:element name="rawValue" type="xsi:string"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>
</xsi:schema>
```

A.2 FAALightSdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="FAALightSdu" type="faaLightSdu"/>

  <xsi:complexType name="faaLightSdu">
    <xsi:all>
```

```
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="computedLightStatus" type="xsi:long"
nillable="true"/>
      <xsi:element name="computedTimeOfDay" type="xsi:long" nillable="true"/>
      <xsi:element name="dipSwitch" type="xsi:int" nillable="true"/>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="intrusionDetection" type="xsi:long"
nillable="true"/>
      <xsi:element name="invalidRmuConfig" type="xsi:int" nillable="true"/>
      <xsi:element name="Failures">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="lightNotFunctional" type="xsi:long"
nillable="true"/>
      <xsi:element name="lightsOutLatch" type="xsi:int" nillable="true"/>
      <xsi:element name="nightTooShort" type="xsi:int" nillable="true"/>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="opCode" type="xsi:long" nillable="true"/>
      <xsi:element name="rmuVersion" type="xsi:long" nillable="true"/>
      <xsi:element name="powerSupplyStatus" type="xsi:long" nillable="true"/>
      <xsi:element name="powerSupplyType" type="xsi:long" nillable="true"/>
      <xsi:element name="pushbuttonStatus" type="xsi:long" nillable="true"/>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="replaceBattery" type="xsi:long" nillable="true"/>
      <xsi:element name="timeOfDay" type="xsi:long" nillable="true"/>
      <xsi:element name="timeOfDayHoursPast" type="xsi:long"
nillable="true"/>
      <xsi:element name="totalBulbs" type="xsi:long" nillable="true"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
```

```
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>
</xsi:schema>
```

A.3 FCISdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="FCISdu" type="fciSdu"/>

  <xsi:complexType name="fciSdu">
    <xsi:all>
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="Version" type="Version"/>
      <xsi:element name="ReportType" type="ReportType"/>
      <xsi:element name="V2ReportType" type="V2ReportType"/>
      <xsi:element name="millis" type="xsi:int"/>
      <xsi:element name="BatteryVoltage" type="xsi:int"/>
      <xsi:element name="MechanicalTargetFaulted" type="xsi:boolean"/>
      <xsi:element name="LowBattery" type="xsi:boolean"/>
      <xsi:element name="Armed" type="xsi:boolean"/>
      <xsi:element name="FlashError" type="xsi:boolean"/>
      <xsi:element name="RamError" type="xsi:boolean"/>
      <xsi:element name="FCIEventLarm" type="xsi:boolean"/>
      <xsi:element name="FirmwareRelease" type="xsi:int"/>
      <xsi:element name="FirmwareReleaseMinor" type="xsi:int"/>
      <xsi:element name="FaultThresholdAmps" type="xsi:int"/>
      <xsi:element name="FaultMagnitudeAmps" type="xsi:int"/>
      <xsi:element name="BatteryRemainingAmpSeconds" type="xsi:int"/>
      <xsi:element name="MomentaryFaultCount" type="xsi:int"/>
      <xsi:element name="MomentaryLoadPickupCount" type="xsi:int"/>
      <xsi:element name="MomentaryLossOfCurrentCount" type="xsi:int"/>
      <xsi:element name="MomentaryLossOfVoltageCount" type="xsi:int"/>
      <xsi:element name="MomentaryFaultDisturbance" type="xsi:int"/>
      <xsi:element name="CumulativeFaultCount" type="xsi:int"/>
      <xsi:element name="CumulativeLossOfCurrentCount" type="xsi:int"/>
      <xsi:element name="CumulativeLossOfVoltageCount" type="xsi:int"/>
      <xsi:element name="CumulativeLossOfCurrentAndVoltageCount"
type="xsi:int"/>
      <xsi:element name="PeakLoadFlag" type="xsi:boolean"/>
      <xsi:element name="StatisticIntervals" type="xsi:int"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="Failures">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="PastHourlyAverageCurrentAmps"
type="PastHourlyAverageCurrentAmps"/>
      <xsi:element name="PastHourlyAverageTemperatureCelsius"
type="PastHourlyAverageTemperatureCelsius"/>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="ReportTimestamp" type="xsi:dateTime"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="PastHourlyAverageCurrentAmps">
    <xsi:sequence>
      <xsi:element name="Value" type="xsi:decimal" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
    <xsi:attribute name="BaseTime" type="xsi:long" use="required"/>
  </xsi:complexType>

  <xsi:complexType name="PastHourlyAverageTemperatureCelsius">
    <xsi:sequence>
      <xsi:element name="Value" type="xsi:decimal" minOccurs="0"
maxOccurs="unbounded"/>
    </xsi:sequence>
    <xsi:attribute name="BaseTime" type="xsi:long" use="required"/>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="Version">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="FCI_VERSION_ONE"/>
      <xsi:enumeration value="FCI_VERSION_TWO"/>
      <xsi:enumeration value="FCI_VERSION_TWO_B"/>
    </xsi:restriction>
  </xsi:simpleType>

  <xsi:simpleType name="ReportType">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="PERIODIC"/>
      <xsi:enumeration value="FAULT"/>
      <xsi:enumeration value="LOSS_OF_CURRENT_FAULT"/>
      <xsi:enumeration value="LOSS_OF_VOLTAGE_FAULT"/>
      <xsi:enumeration value="LOSS_OF_CURRENT_AND_VOLTAGE_FAULT"/>
      <xsi:enumeration value="TIMEOUT_INTERVAL_EXCEEDED"/>
    </xsi:restriction>
  </xsi:simpleType>

  <xsi:simpleType name="V2ReportType">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="NULL"/>
      <xsi:enumeration value="Deployment"/>
      <xsi:enumeration value="Periodic"/>
      <xsi:enumeration value="Restoration"/>
      <xsi:enumeration value="Fault"/>
      <xsi:enumeration value="LOSS_OF_CURRENT"/>
      <xsi:enumeration value="LOSS_OF_VOLTAGE"/>
      <xsi:enumeration value="LOSS_OF_CURRENT_VOLTAGE"/>
      <xsi:enumeration value="DOWNLINK_RESPONSE_ACK"/>
      <xsi:enumeration value="DONWLINK_RESPONSE_NEGATIVE_ACK"/>
      <xsi:enumeration value="FCI_SOFTWARE_UPGRADE_ACK"/>
      <xsi:enumeration value="FCI_SOFTWARE_UPGRADE_NEGATIVE_ACK"/>
      <xsi:enumeration value="MOMENTARY_FAULT"/>
      <xsi:enumeration value="MOMENTARY_LOSS_OF_CURRENT"/>
      <xsi:enumeration value="MOMENTARY_DISTURBANCE_FAULT"/>
      <xsi:enumeration value="MOMENTARY_LOAD_PICKUP_EVENT"/>
      <xsi:enumeration value="TIMEOUT_INTERVAL_EXCEEDED"/>
      <xsi:enumeration value="Invalid"/>
    </xsi:restriction>
  </xsi:simpleType>
</xsi:schema>
```

A.4 KONWPTSdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="KONWPTSdu" type="konwptSdu"/>

  <xsi:complexType name="konwptSdu">
    <xsi:all>
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="batteryHealth" type="xsi:int"/>
      <xsi:element name="deviceTimestamp" type="xsi:long" minOccurs="0"/>
      <xsi:element name="deviceType" type="xsi:int"/>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="invalid" type="xsi:boolean"/>
      <xsi:element name="Failures">
```

```
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="lowBatteryAlarm" type="xsi:boolean"/>
      <xsi:element name="messageType" type="messageType"/>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="overThresholdAlarm" type="xsi:boolean"/>
      <xsi:element name="pressure" type="xsi:double"/>
      <xsi:element name="rateOfChangeAlarm" type="xsi:boolean"/>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="sensorFaultAlarm" type="xsi:boolean"/>
      <xsi:element name="tempOverThresholdAlarm" type="xsi:boolean"/>
      <xsi:element name="timeAcquisitionError" type="xsi:boolean"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="underThresholdAlarm" type="xsi:boolean"/>
      <xsi:element name="voltage" type="xsi:double"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="messageType">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="Periodic"/>
      <xsi:enumeration value="GetDeviceSettings"/>
      <xsi:enumeration value="SetDeviceSettings"/>
      <xsi:enumeration value="GetDeviceStats"/>
      <xsi:enumeration value="ResetAlarms"/>
    </xsi:restriction>
  </xsi:simpleType>
</xsi:schema>
```

A.5 SEL8301Sdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="SEL8301ASdu" type="sel8301ASdu"/>

  <xsi:complexType name="sel8301ASdu">
    <xsi:all>
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="batteryVoltage" type="xsi:int" nillable="true"/>
      <xsi:element name="deviceEnvCurrent" type="xsi:int"/>
      <xsi:element name="deviceEnvTemperature" type="xsi:int"/>
      <xsi:element name="deviceExternalPower" type="xsi:boolean"/>
      <xsi:element name="deviceFlashError" type="xsi:boolean"/>
      <xsi:element name="deviceID" type="xsi:long" nillable="true"/>
      <xsi:element name="deviceLowBattery" type="xsi:boolean"/>
      <xsi:element name="deviceOutageWarning" type="xsi:boolean"/>
      <xsi:element name="devicePortError" type="xsi:boolean"/>
      <xsi:element name="devicePortFailure" type="xsi:boolean"/>
      <xsi:element name="deviceRamError" type="xsi:boolean"/>
      <xsi:element name="deviceUnderpowered" type="xsi:boolean"/>
      <xsi:element name="downlinkHex" type="xsi:string"/>
      <xsi:element name="downlinkMessageLength" type="xsi:int"/>
      <xsi:element name="downlinkReserved" type="xsi:int"/>
      <xsi:element name="firmwareVersion" type="xsi:string" nillable="true"/>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="Failures">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="offlinePorts" type="xsi:boolean" nillable="true"/>
      <xsi:element name="packetLength" type="xsi:int" nillable="true"/>
      <xsi:element name="packetType" type="xsi:int" nillable="true"/>
      <xsi:element name="payloadPacketCount" type="xsi:int" nillable="true"/>
      <xsi:element name="portFirmwareVersion" type="xsi:string"
nillable="true"/>
      <xsi:element name="Ports">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Port" type="SEL8301APortSdu" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="reportTimestamp" type="xsi:long" nillable="true"/>
      <xsi:element name="statisticsInterval" type="xsi:int"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="SEL8301APortSdu">
    <xsi:all>
      <xsi:element name="armed" type="xsi:boolean"/>
      <xsi:element name="attachedSensorField" type="xsi:int"/>
      <xsi:element name="cumulativeFaults" type="xsi:int"/>
      <xsi:element name="cumulativeLossOfCurrent" type="xsi:int"/>
      <xsi:element name="cumulativeReserved0" type="xsi:int"/>
      <xsi:element name="cumulativeReserved1" type="xsi:int"/>
      <xsi:element name="eventOffset" type="xsi:int"/>
      <xsi:element name="fault" type="xsi:boolean"/>
      <xsi:element name="faultInfoCurrent" type="xsi:int"/>
      <xsi:element name="faultThreshold" type="xsi:int"/>
      <xsi:element name="id" type="xsi:long"/>
      <xsi:element name="loadraw" type="xsi:string"/>
      <xsi:element name="loc" type="xsi:boolean"/>
      <xsi:element name="momentaryDisturbanceFault" type="xsi:int"/>
      <xsi:element name="momentaryFaults" type="xsi:int"/>
      <xsi:element name="momentaryLoadPickUp" type="xsi:int"/>
      <xsi:element name="momentaryLossOfCurrent" type="xsi:int"/>
      <xsi:element name="offline" type="xsi:boolean"/>
      <xsi:element name="isPeak" type="xsi:boolean"/>
      <xsi:element name="portFailure" type="xsi:boolean"/>
      <xsi:element name="portId" type="xsi:int"/>
      <xsi:element name="portRestoration" type="xsi:boolean"/>
      <xsi:element name="powerDirection" type="xsi:boolean"/>
      <xsi:element name="reportTimestamp" type="xsi:long"/>
      <xsi:element name="nodeId" type="xsi:long"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>
</xsi:schema>
```

A.6 TIQSdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="TIQSdu" type="tiqSdu"/>

  <xsi:complexType name="tiqSdu">
    <xsi:all>
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="Alarms" minOccurs="0">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Alarm" type="Alarm" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="AmbientTemperature" type="xsi:int" nillable="true"/>
      <xsi:element name="AmpsWindings1" type="xsi:int" nillable="true"/>
      <xsi:element name="AmpsWindings2" type="xsi:int" nillable="true"/>
      <xsi:element name="AmpsWindings3" type="xsi:int" nillable="true"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="Conductor1Temperature" type="xsi:int"
nillable="true"/>
      <xsi:element name="Conductor2Temperature" type="xsi:int"
nillable="true"/>
      <xsi:element name="Conductor3Temperature" type="xsi:int"
nillable="true"/>
      <xsi:element name="FaultDurationCyclesWinding1" type="xsi:int"
nillable="true"/>
      <xsi:element name="FaultDurationCyclesWinding2" type="xsi:int"
nillable="true"/>
      <xsi:element name="FaultDurationCyclesWinding3" type="xsi:int"
nillable="true"/>
      <xsi:element name="Message_Id" type="xsi:long"/>
      <xsi:element name="Failures">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="MaxLoadPercentageWindings1" type="xsi:double"
nillable="true"/>
      <xsi:element name="MaxLoadPercentageWindings2" type="xsi:double"
nillable="true"/>
      <xsi:element name="MaxLoadPercentageWindings3" type="xsi:double"
nillable="true"/>
      <xsi:element name="MaxVoltageSwellWinding1" type="xsi:int"
nillable="true"/>
```

```
      <xsi:element name="MaxVoltageSwellWinding2" type="xsi:int"
nillable="true"/>
      <xsi:element name="MaxVoltageSwellWinding3" type="xsi:int"
nillable="true"/>
      <xsi:element name="MinVoltageSagWinding1" type="xsi:int"
nillable="true"/>
      <xsi:element name="MinVoltageSagWinding2" type="xsi:int"
nillable="true"/>
      <xsi:element name="MinVoltageSagWinding3" type="xsi:int"
nillable="true"/>
      <xsi:element name="NeutralAmps" type="xsi:int" nillable="true"/>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="PhaseAngleWindings1" type="xsi:int"
nillable="true"/>
      <xsi:element name="PhaseAngleWindings2" type="xsi:int"
nillable="true"/>
      <xsi:element name="PhaseAngleWindings3" type="xsi:int"
nillable="true"/>
      <xsi:element name="PowerFactorWindings1" type="xsi:double"
nillable="true"/>
      <xsi:element name="PowerFactorWindings2" type="xsi:double"
nillable="true"/>
      <xsi:element name="PowerFactorWindings3" type="xsi:double"
nillable="true"/>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="TopOil1Temperature" type="xsi:int" nillable="true"/>
      <xsi:element name="TopOil2Temperature" type="xsi:int" nillable="true"/>
      <xsi:element name="TopOil3Temperature" type="xsi:int" nillable="true"/>
      <xsi:element name="Transformer_Id" type="xsi:int"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="AccumulatedPowerKVARWinding1" type="xsi:double"
nillable="true"/>
      <xsi:element name="AccumulatedPowerKVARWinding2" type="xsi:double"
nillable="true"/>
      <xsi:element name="AccumulatedPowerKVARWinding3" type="xsi:double"
nillable="true"/>
      <xsi:element name="AccumulatedPowerKWHRWinding1" type="xsi:double"
nillable="true"/>
      <xsi:element name="AccumulatedPowerKWHRWinding2" type="xsi:double"
nillable="true"/>
      <xsi:element name="AccumulatedPowerKWHRWinding3" type="xsi:double"
nillable="true"/>
      <xsi:element name="Winding1FaultMagnitude" type="xsi:int"
nillable="true"/>
      <xsi:element name="Winding1HotSpot" type="xsi:int" nillable="true"/>
      <xsi:element name="Winding2FaultMagnitude" type="xsi:int"
nillable="true"/>
      <xsi:element name="Winding2HotSpot" type="xsi:int" nillable="true"/>
      <xsi:element name="Winding3FaultMagnitude" type="xsi:int"
nillable="true"/>
      <xsi:element name="Winding3HotSpot" type="xsi:int" nillable="true"/>
      <xsi:element name="Windings1FltMagExceed" type="xsi:boolean"
nillable="true"/>
      <xsi:element name="Windings1Voltage" type="xsi:int" nillable="true"/>
      <xsi:element name="Windings2FltMagExceed" type="xsi:boolean"
nillable="true"/>
```

```
      <xsi:element name="Windings2Voltage" type="xsi:int" nillable="true"/>
      <xsi:element name="Windings3FltMagExceed" type="xsi:boolean"
nillable="true"/>
      <xsi:element name="Windings3Voltage" type="xsi:int" nillable="true"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="Alarm">
    <xsi:sequence>
      <xsi:element name="Failure" type="xsi:string"/>
      <xsi:element name="Threshold" type="xsi:int"/>
      <xsi:element name="Winding" type="xsi:int"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>
</xsi:schema>
```

A.7 ElectricAMISdu Schema

```
<?xml version="1.0" standalone="yes"?>
<xsi:schema elementFormDefault="unqualified" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">

  <xsi:element name="ElectricAMIEventsSdu" type="electricAMIEventsSdu"/>

  <xsi:element name="ElectricAMIReadingsSdu" type="electricAMIReadingsSdu"/>

  <xsi:element name="ElectricAMISdu" type="electricAMISdu"/>

  <xsi:complexType name="electricAMISdu">
    <xsi:all>
      <xsi:element name="AppTypeID" type="xsi:int"/>
      <xsi:element name="App_Device_Id" type="xsi:long"/>
      <xsi:element name="eventData">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="event" type="electricAMIEventsSdu"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="Failures">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="item" type="FailureType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="messageSummary" type="xsi:string" minOccurs="0"/>
      <xsi:element name="Device_Id" type="xsi:long"/>
      <xsi:element name="readingData">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="reading" type="electricAMIReadingsSdu"
minOccurs="0" maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
      <xsi:element name="Message_Received" type="xsi:long"/>
      <xsi:element name="ULP_Node_Id" type="xsi:long"/>
      <xsi:element name="UplinkType" type="UplinkType" minOccurs="0"/>
      <xsi:element name="MetaAttribs">
        <xsi:complexType>
          <xsi:sequence>
            <xsi:element name="Attrib" type="XMLMetaAttrib" minOccurs="0"
maxOccurs="unbounded"/>
          </xsi:sequence>
        </xsi:complexType>
      </xsi:element>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="electricAMIEventsSdu">
    <xsi:all>
      <xsi:element name="amiSduId" type="xsi:long"/>
      <xsi:element name="appDeviceId" type="xsi:long"/>
      <xsi:element name="categoryCodeId" type="xsi:string" minOccurs="0"/>
      <xsi:element name="eventDescription" type="xsi:string" minOccurs="0"/>
      <xsi:element name="failureTypeId" type="xsi:int" minOccurs="0"/>
      <xsi:element name="occurenceDate" type="xsi:dateTime" minOccurs="0"/>
      <xsi:element name="Received" type="xsi:dateTime" minOccurs="0"/>
      <xsi:element name="ulpNodeId" type="xsi:long"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="FailureType">
    <xsi:sequence>
      <xsi:element name="Id" type="xsi:int"/>
      <xsi:element name="Severity" type="xsi:int" nillable="true"/>
      <xsi:element name="Time" type="xsi:long" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:complexType name="electricAMIReadingsSdu">
    <xsi:all>
      <xsi:element name="amiSduId" type="xsi:long"/>
      <xsi:element name="appDeviceId" type="xsi:long"/>
      <xsi:element name="delta" type="xsi:double" minOccurs="0"/>
      <xsi:element name="qualityCode" type="xsi:string" minOccurs="0"/>
      <xsi:element name="qualityCodeDescription" type="xsi:string" 
minOccurs="0"/>
      <xsi:element name="readingDate" type="xsi:dateTime" minOccurs="0"/>
      <xsi:element name="readingTypeId" type="xsi:string" minOccurs="0"/>
      <xsi:element name="readingTypeDescription" type="xsi:string" 
minOccurs="0"/>
      <xsi:element name="Received" type="xsi:dateTime" minOccurs="0"/>
      <xsi:element name="ulpNodeId" type="xsi:long"/>
      <xsi:element name="value" type="xsi:double"/>
    </xsi:all>
  </xsi:complexType>

  <xsi:complexType name="XMLMetaAttrib">
    <xsi:sequence>
      <xsi:element name="Name" type="xsi:string" nillable="true"/>
      <xsi:element name="Value" type="xsi:string" nillable="true"/>
    </xsi:sequence>
  </xsi:complexType>

  <xsi:simpleType name="UplinkType">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="NONE"/>
      <xsi:enumeration value="EVENT"/>
      <xsi:enumeration value="BULK_READINGS"/>
      <xsi:enumeration value="INTERVAL_READINGS"/>
      <xsi:enumeration value="METER_INFO"/>
      <xsi:enumeration value="RELAY_STATE"/>
    </xsi:restriction>
  </xsi:simpleType>
</xsi:schema>
```

Appendix B Abbreviations and Terms

Abbreviation/Term	Definition
OTV	Critical Infrastructure Monitoring Application. The network component that passes data from the Gateway to the associated upstream databases.
FAA	Federal Aviation Administration
FCI	Fault Circuit Indicator
HTTP	Hypertext Transfer Protocol
HTTP GET	Requests a representation of the specified resource. Requests using GET should only retrieve data and should have no other effect.
HTTP POST	Requests that the server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI. The data POSTed might be, for example: ■ An annotation for existing resources ■ A message for a bulletin board, newsgroup, mailing list, or comment thread ■ A block of data that is the result of submitting a web form to a data-handling process ■ An item to add to a database
KONWPT	The wireless pressure transmitter developed by KONČAR that is part of the wireless pipeline pressure monitoring system. The KONWPT is a battery powered autonomous unit for pipeline pressure measurement.
Node	The wireless module developed by On-Ramp Wireless that integrates with OEM sensors and communicates sensor data to an Access Point. Also, the generic term used interchangeably with endpoint device.
Payload	The XML payload is the body data or contents (e.g., the “cargo”) of a data transmission.
REST	Representational State Transfer
SDU	Service Data Unit
SEL	Schweitzer Engineering Laboratories
SSH	Secure Shell
SSL	Secure Socket Layer
TIQ	TransformerIQ
URL	Uniform Resource Locator
XML	Extensible Markup Language