ELECTRIC POWER RESEARCH INSTITUTE

Outage Data Initiative

Interface and Test Script Overview

Daniel Lowe
Engineer/Scientist I
Enterprise Architecture and Integration

7/12/2019

What is ODIN?

OAK RIDGE National Laboratory

EPRI ELECTRIC POWER RESEARCH INSTITUTE

U.S. DEPARTMENT OF ENERGY

ODIN Status

- Four official participants (OMS vendor – NISC)
  - Mason PUD 3
  - Clallam County PUD
  - Orcas Power & Light Co-Op (OPALCO)
  - Douglas PUD
- One utility in-process of joining: Benton PUD
- Receiving direct outage data feeds from 3 official participants at zip code resolution through their OMS vendor, NISC

Testing Resources

- Test Script
  - Test cases – documented expectations
- Test Cases
  - Leveraging SoapUI
- Demonstrate!
  - PlugFest –
  - EPRI, Department of Energy, Duke
  - September 24 – 26th
  - EPRI Charlotte campus
  - Registration will open soon

Outage Data Use Cases for Exchanging Data

- UC 1: Within a utility (Detailed outage data)
- UC 2: Between utilities (Generalized outage data)
- UC 3: Between utilities and mutual aid organizations: Regional Power Outage Dashboard for enhanced situational awareness (Could be General or Detailed)
- UC 4: With state emergency management agencies (Detailed outage data)
- UC 5: With local Police/Fire/EMS (Detailed outage data)
- UC 6: With media/public (Utility public outage map & external CIM Interface)
- UC 7: With "Smart" municipalities/cities (Customization of UC 6)
- UC 8: With other infrastructure providers (e.g. Cellular, Joint utility providers)

- Potential Use Cases for Continued Research:
- UC 9: Integration of Social Media for outage reporting to/from customers
- UC 10: Integration of Augmented Reality, Drones, mobile platforms and WMS with GIS & OMS for damage assessment
- UC 11: Extend outage messaging to AMI meters
- UC 12: Improved Post-Storm Analysis "report card", analysis of outage reporting codes

Use Case 6 is primary focus for this Demonstration

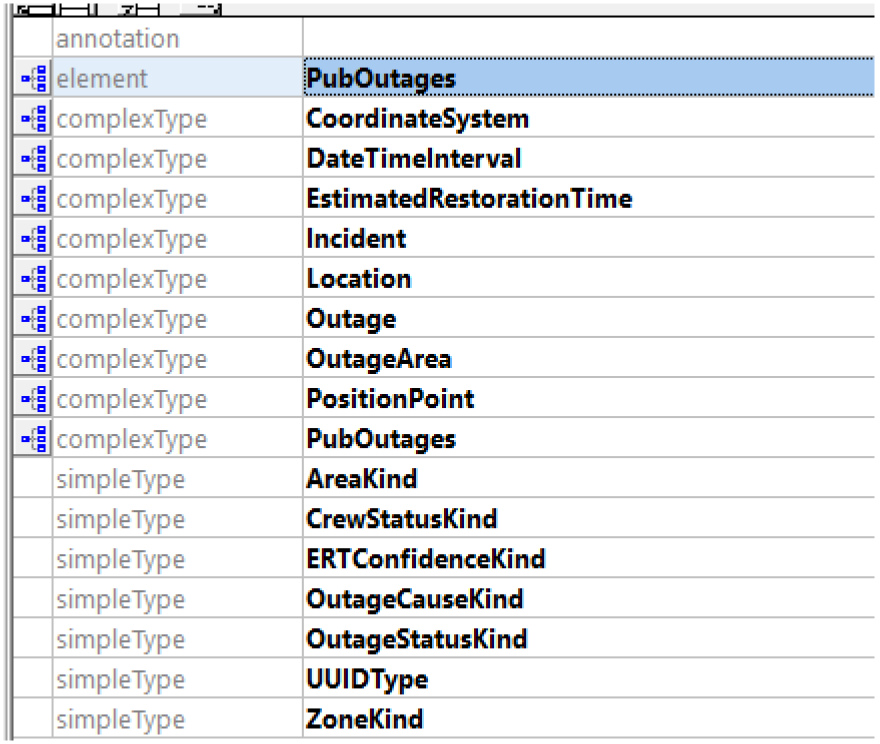
Seattle City Light

- The first utility to get onboard with ODI
- Implemented the translation to the standard in less than 10 hours
- Provided opportunity to share their outage data with surrounding utilities

Schema Overview

PubOutages Schema

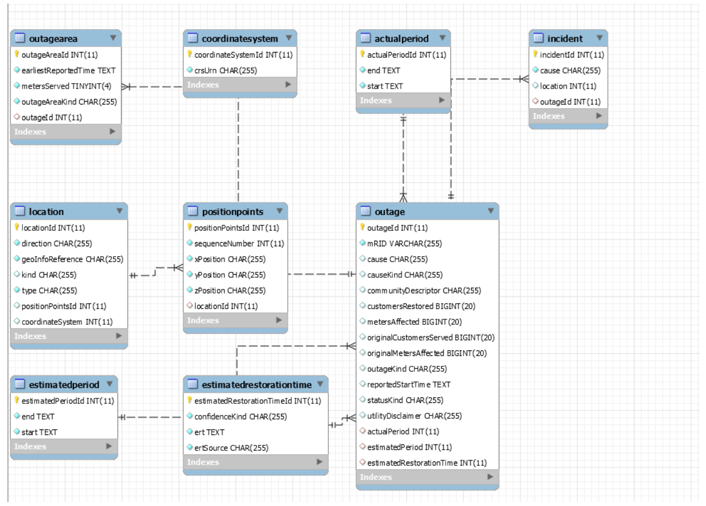
- Schema can be standalone
- All complex types are optional
- For database storage…
  - Just need one table at minimum OR
  - One table per complex type being used for systems that want to share more data

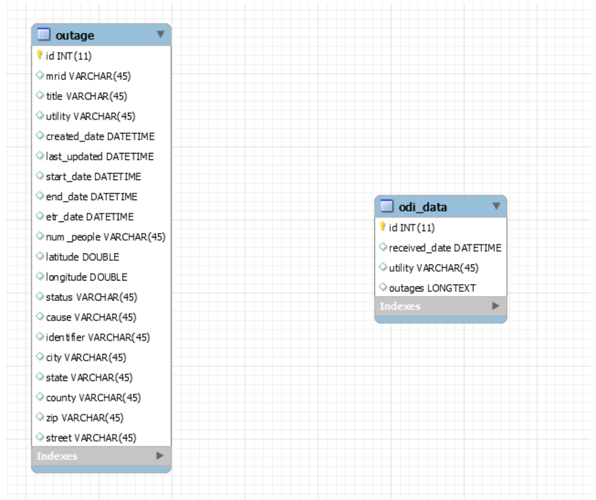


annotation	
element	PubOutages
complexType	CoordinateSystem
complexType	DateTimeInterval
complexType	EstimatedRestorationTime
complexType	Incident
complexType	Location
complexType	Outage
complexType	OutageArea
complexType	PositionPoint
complexType	PubOutages
simpleType	AreaKind
simpleType	CrewStatusKind
simpleType	ERTConfidenceKind
simpleType	OutageCauseKind
simpleType	OutageStatusKind
simpleType	UUIDType
simpleType	ZoneKind

PubOutages Schema Database Representation

- Optional Elements in the schema allow for either complicated or simple database designs



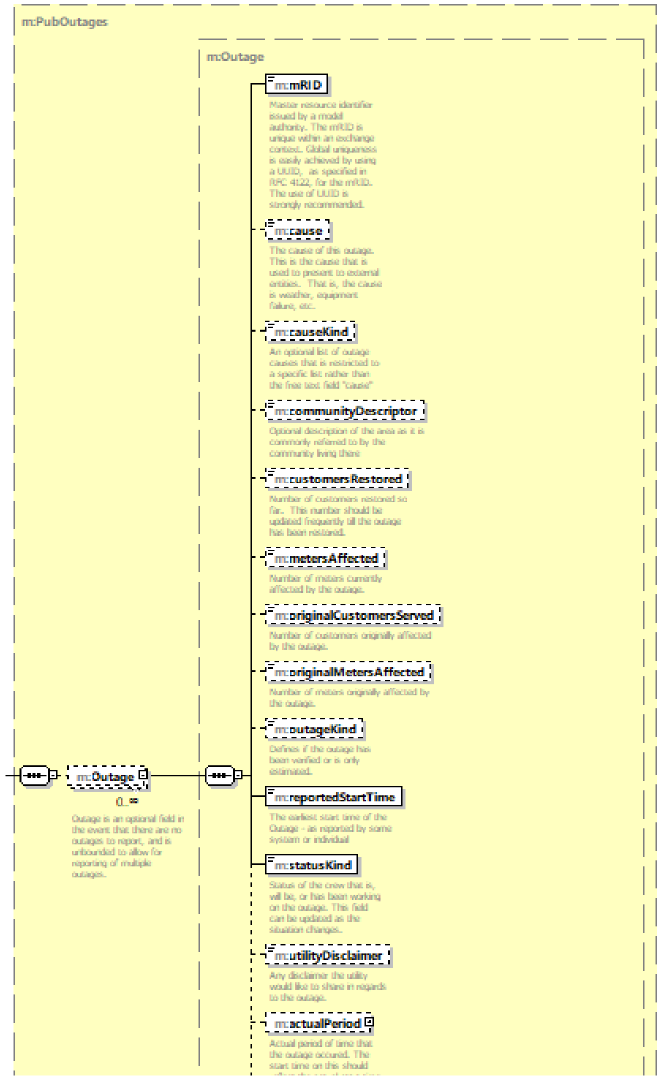


- Both types of setups can accomodate ODI standard

PubOutages.xsd Schema Changes

- **mRID**, **reportedStartTime**, and **statusKind** are now <u>required</u>

- **mRID**: Now enforced as UUID-type
- **reportedStartTime**: time that the outage started as reported by system or individual
- **statusKind:** Status of the crew that is or will be working on the outage



Outage is an optional field in the event that there are no outages to report, and is unbounded to allow for reporting of multiple outages.

m:mRID: Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.

m:cause: The cause of this outage. This is the cause that is used to present to external entities. That is, the cause is weather, equipment failure, etc.

m:causeKind: An optional list of outage causes that is restricted to a specific list rather than the free text field "cause"

m:communityDescriptor: Optional description of the area as it is commonly referred to by the community living there

m:customersRestored: Number of customers restored so far. This number should be updated frequently till the outage has been restored.

m:metersAffected: Number of meters currently affected by the outage.

m:originalCustomersServed: Number of customers originally affected by the outage.

m:originalMetersAffected: Number of meters originally affected by the outage.

m:outageKind: Defines if the outage has been verified or is only estimated.

m:reportedStartTime: The earliest start time of the Outage - as reported by some system or individual

m:statusKind: Status of the crew that is, will be, or has been working on the outage. This field can be updated as the situation changes.

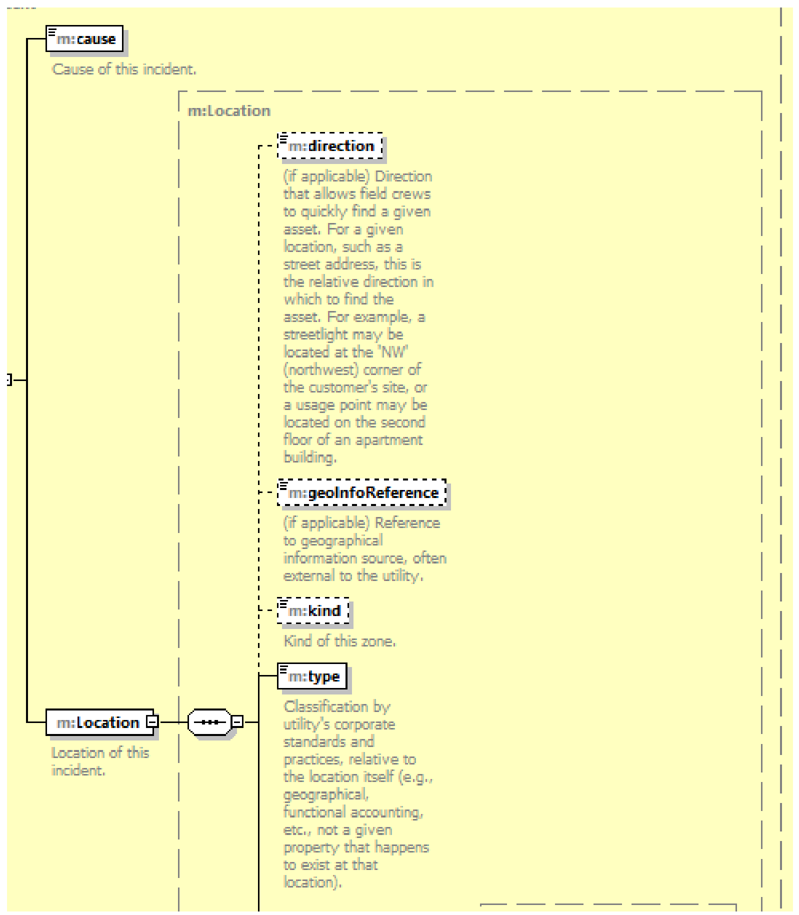
m:utilityDisclaimer: Any disclaimer the utility would like to share in regards to the outage.

m:actualPeriod: Actual period of time that the outage occurred. The start time on this should

PubOutages.xsd Schema Changes

- **Location: direction**, **kind**, **geoInfoReference**, and **zPosition** are now optional

- Previous versions of the schema described some fields as if they were optional but still made them mandatory
- It is unlikely that many more fields will be made optional at this point
- Changes will be proposed to IEC 4th Qtr 2019



Cause of this incident.

(if applicable) Direction that allows field crews to quickly find a given asset. For a given location, such as a street address, this is the relative direction in which to find the asset. For example, a streetlight may be located at the 'NW' (northwest) corner of the customer's site, or a usage point may be located on the second floor of an apartment building.

(if applicable) Reference to geographical information source, often external to the utility.

Kind of this zone.

Classification by utility's corporate standards and practices, relative to the location itself (e.g., geographical, functional accounting, etc., not a given property that happens to exist at that location).

Location of this incident.

Schema Changes Recap

- **mRID** enforced as UUID type
- **reportedStartTime** and **statusKind** now required
- **Location.direction**, **Location.kind**, **Location. geoInfoReference**, and **PositionPoints.zPosition** now optional
- Other mandatory elements nested within optional elements have stayed the same

ODI Test Script for Integration

SOAP vs REST Web Services

- SOAP
  - Simple Object Access Protocol
  - Message Protocol for sending XML messages over HTTP
  - Allows for stricter API
  - Requires more computing power/bandwidth
- REST
  - REpresentational State Transfer
  - Architecture Style (not a protocol like SOAP)
  - Messages aren’t restricted to XML, could also be plain text, JSON, HTML, etc.
  - More lightweight

- Current ODI implementations are in REST
- Test Script is currently written for REST implementations
- SOAP implementation guidance will be provided at a future date
- EPRI’s Test Harness will eventually support both for ODI

ODI Test Script

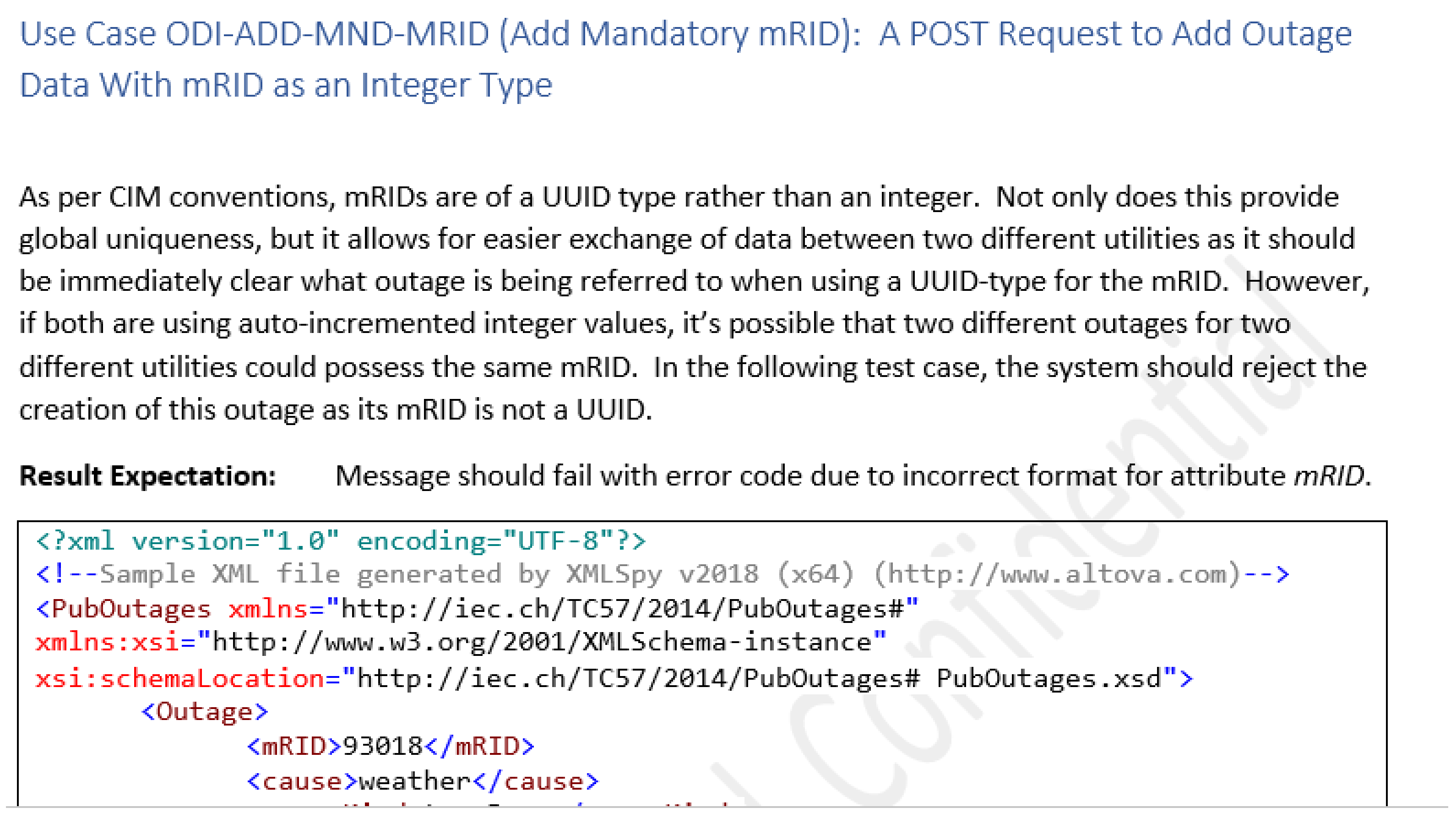
- EPRI’s ODI Test Script describes 27 total use cases
  - POST: 16
  - PUT: 7
  - PATCH: 1
  - GET: 2
  - DELETE: 1
- Test Script was written for RESTful implementation
- A SOAP version may be created if there’s enough demand
- Not all use cases must be passed to be considered compliant

ODI Test Script Contents

- Test Script provides instructions on how to be used with SoapUI
  - A SoapUI file is provided with all use case messages preloaded
    - Endpoint URL will need to be changed to match service being tested
- Test Script describes the PubOutages schema and its various complex types
- Each use case follows a specific naming convention
- Expected results are posted for each use case

Test Script Example Use Case - POST

- Test for correct usage of **mRID**



Use Case ODI-ADD-MND-MRID (Add Mandatory mRID): A POST Request to Add Outage Data With mRID as an Integer Type

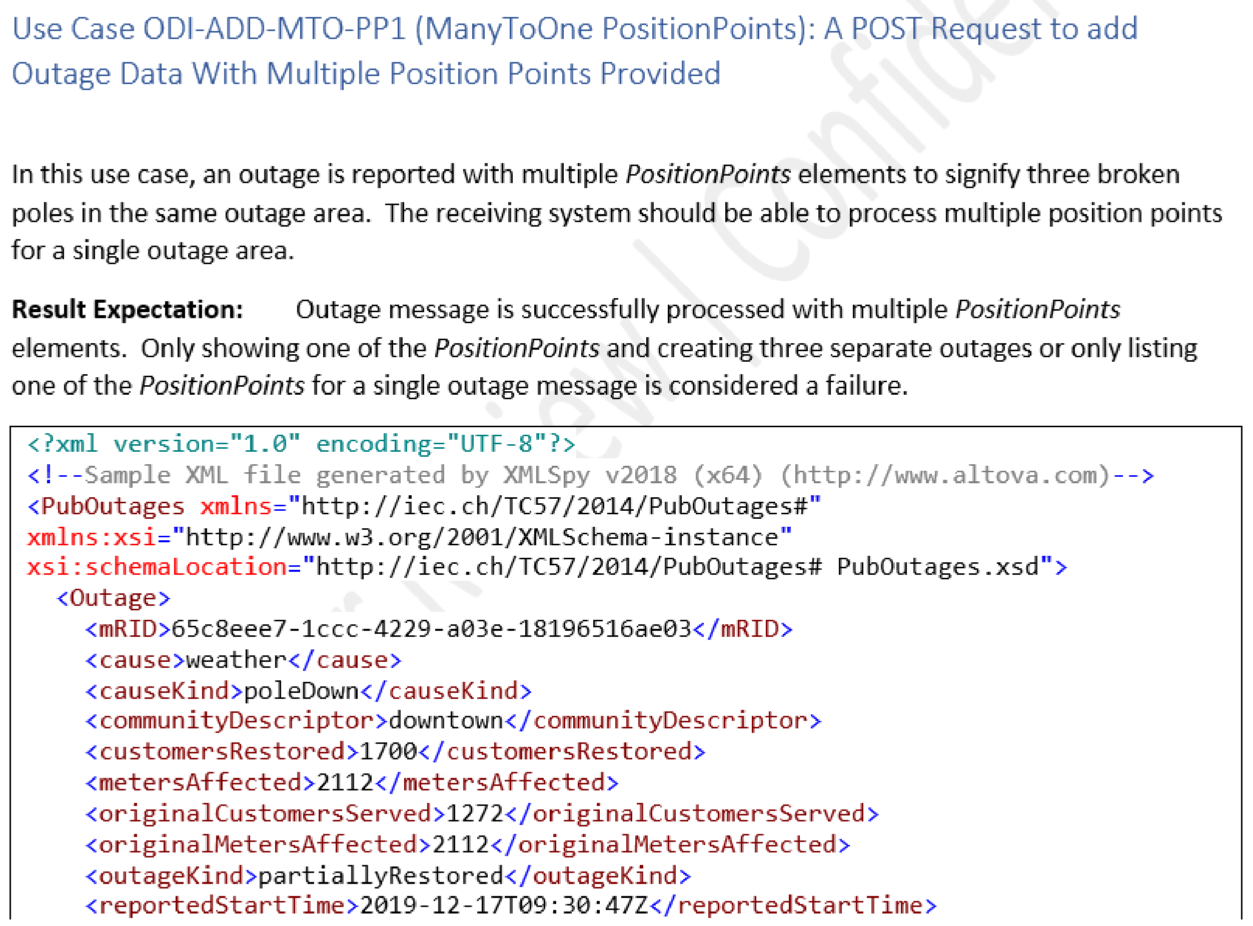
As per CIM conventions, mRIDs are of a UUID type rather than an integer. Not only does this provide global uniqueness, but it allows for easier exchange of data between two different utilities as it should be immediately clear what outage is being referred to when using a UUID-type for the mRID. However, if both are using auto-incremented integer values, it's possible that two different outages for two different utilities could possess the same mRID. In the following test case, the system should reject the creation of this outage as its mRID is not a UUID.

Result Expectation: Message should fail with error code due to incorrect format for attribute *mRID*.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2018 (x64) (http://www.altova.com)-->
<PubOutages xmlns="http://iec.ch/TC57/2014/PubOutages#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2014/PubOutages# PubOutages.xsd">
	<Outage>
		<mRID>93018</mRID>
		<cause>weather</cause>
```

Test Script Example Use Case - POST

- One use case tests for multiple position points for a single outage



Use Case ODI-ADD-MTO-PP1 (ManyToOne PositionPoints): A POST Request to add Outage Data With Multiple Position Points Provided

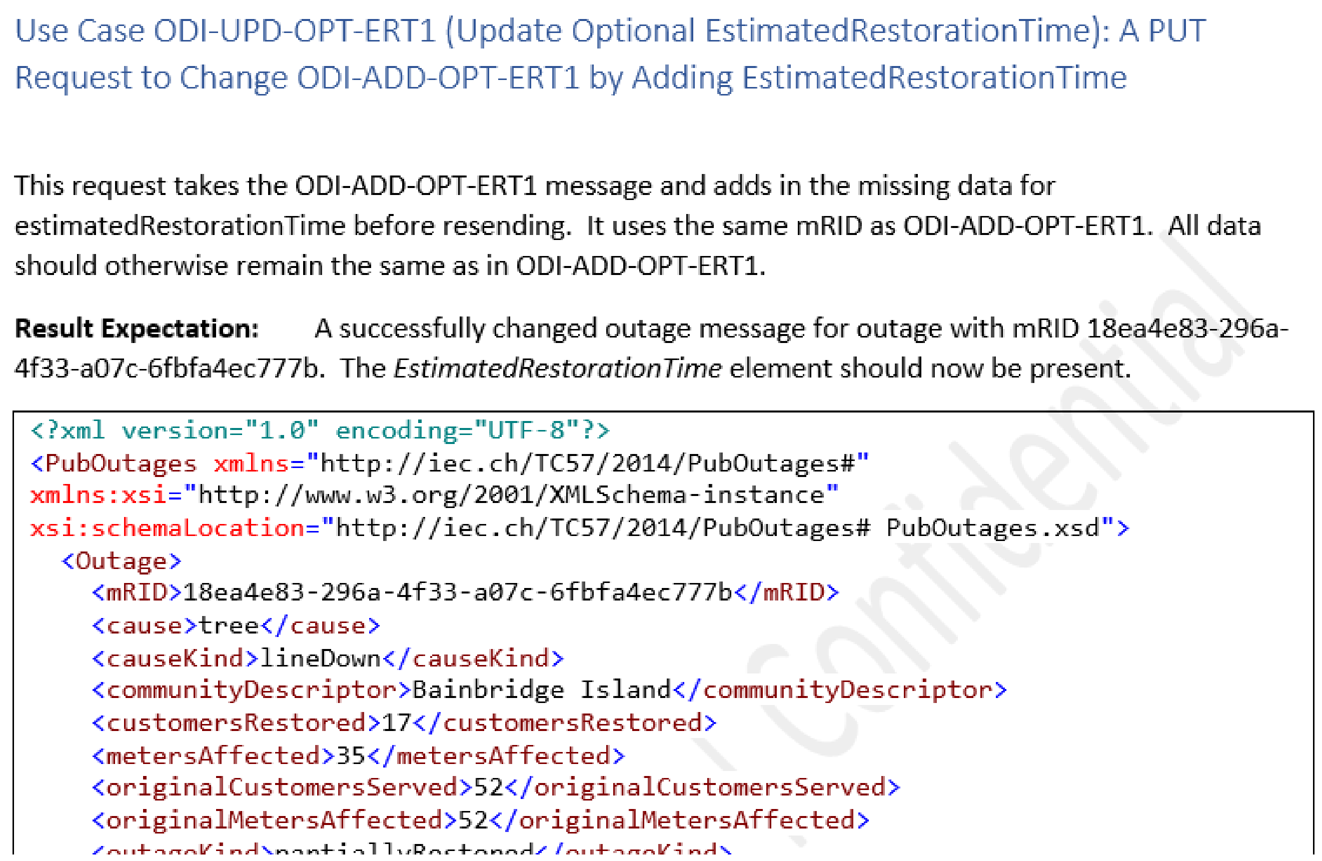
In this use case, an outage is reported with multiple *PositionPoints* elements to signify three broken poles in the same outage area. The receiving system should be able to process multiple position points for a single outage area.

Result Expectation: Outage message is successfully processed with multiple *PositionPoints* elements. Only showing one of the *PositionPoints* and creating three separate outages or only listing one of the *PositionPoints* for a single outage message is considered a failure.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2018 (x64) (http://www.altova.com)-->
<PubOutages xmlns="http://iec.ch/TC57/2014/PubOutages#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2014/PubOutages# PubOutages.xsd">
  <Outage>
    <mRID>65c8eee7-1ccc-4229-a03e-18196516ae03</mRID>
    <cause>weather</cause>
    <causeKind>poleDown</causeKind>
    <communityDescriptor>downtown</communityDescriptor>
    <customersRestored>1700</customersRestored>
    <metersAffected>2112</metersAffected>
    <originalCustomersServed>1272</originalCustomersServed>
    <originalMetersAffected>2112</originalMetersAffected>
    <outageKind>partiallyRestored</outageKind>
    <reportedStartTime>2019-12-17T09:30:47Z</reportedStartTime>
```

Test Script Example Use Case – PUT

- Use case changes existing message to add new data



Use Case ODI-UPD-OPT-ERT1 (Update Optional EstimatedRestorationTime): A PUT Request to Change ODI-ADD-OPT-ERT1 by Adding EstimatedRestorationTime

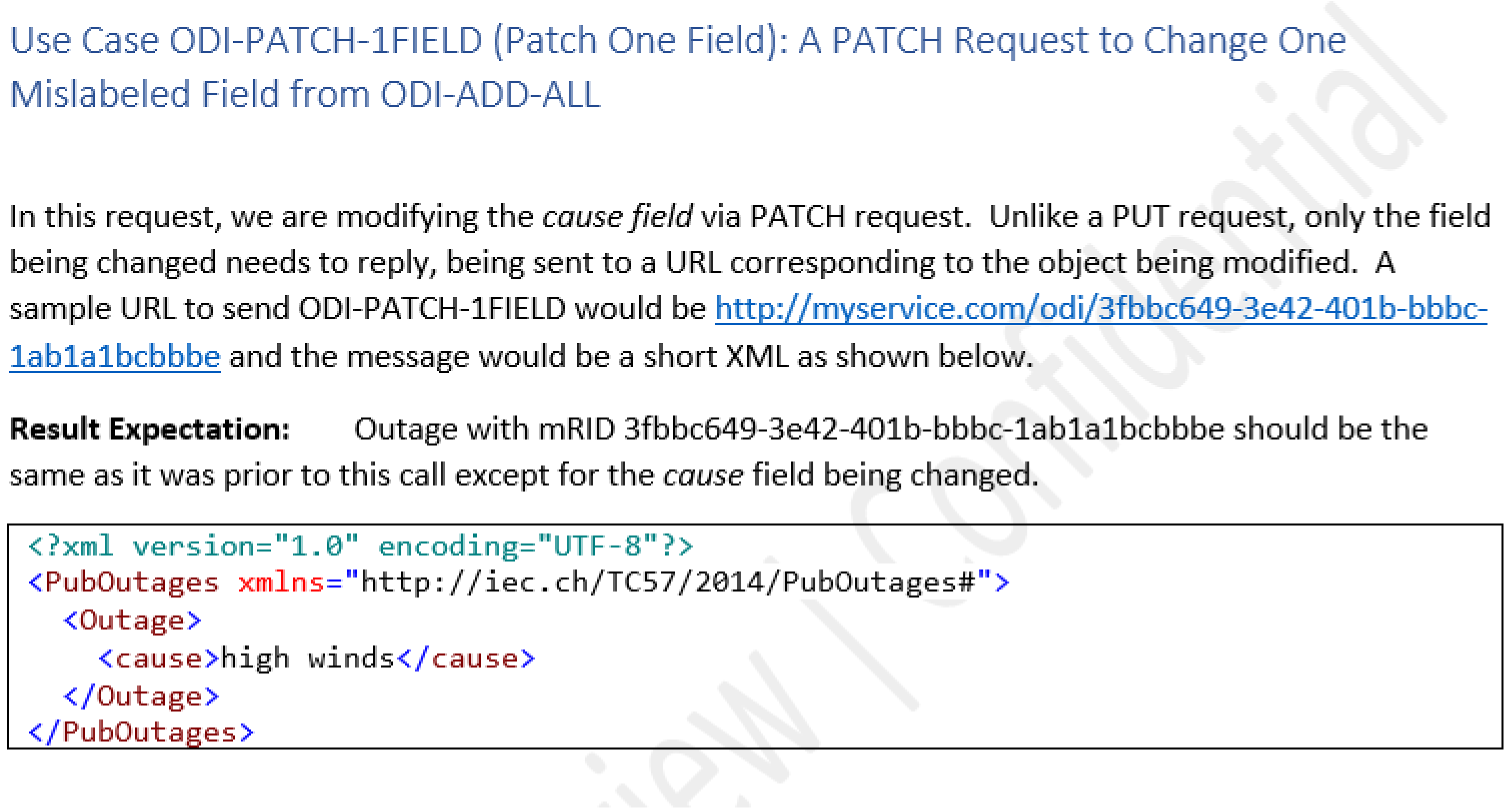
This request takes the ODI-ADD-OPT-ERT1 message and adds in the missing data for estimatedRestorationTime before resending. It uses the same mRID as ODI-ADD-OPT-ERT1. All data should otherwise remain the same as in ODI-ADD-OPT-ERT1.

Result Expectation: A successfully changed outage message for outage with mRID 18ea4e83-296a-4f33-a07c-6fbfa4ec777b. The *EstimatedRestorationTime* element should now be present.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<PubOutages xmlns="http://iec.ch/TC57/2014/PubOutages#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2014/PubOutages# PubOutages.xsd">
  <Outage>
    <mRID>18ea4e83-296a-4f33-a07c-6fbfa4ec777b</mRID>
    <cause>tree</cause>
    <causeKind>lineDown</causeKind>
    <communityDescriptor>Bainbridge Island</communityDescriptor>
    <customersRestored>17</customersRestored>
    <metersAffected>35</metersAffected>
    <originalCustomersServed>52</originalCustomersServed>
    <originalMetersAffected>52</originalMetersAffected>
    <outageKind>partiallyRestored</outageKind>
```

Test Script Example Use Case – PATCH

- Use case demonstrating a PATCH request to update an outage



Use Case ODI-PATCH-1FIELD (Patch One Field): A PATCH Request to Change One Mislabeled Field from ODI-ADD-ALL

In this request, we are modifying the *cause field* via PATCH request. Unlike a PUT request, only the field being changed needs to reply, being sent to a URL corresponding to the object being modified. A sample URL to send ODI-PATCH-1FIELD would be http://myservice.com/odi/3fbbc649-3e42-401b-bbbc-1ab1a1bcbbbe and the message would be a short XML as shown below.

Result Expectation: Outage with mRID 3fbbc649-3e42-401b-bbbc-1ab1a1bcbbbe should be the same as it was prior to this call except for the *cause* field being changed.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<PubOutages xmlns="http://iec.ch/TC57/2014/PubOutages#">
  <Outage>
    <cause>high winds</cause>
  </Outage>
</PubOutages>
```

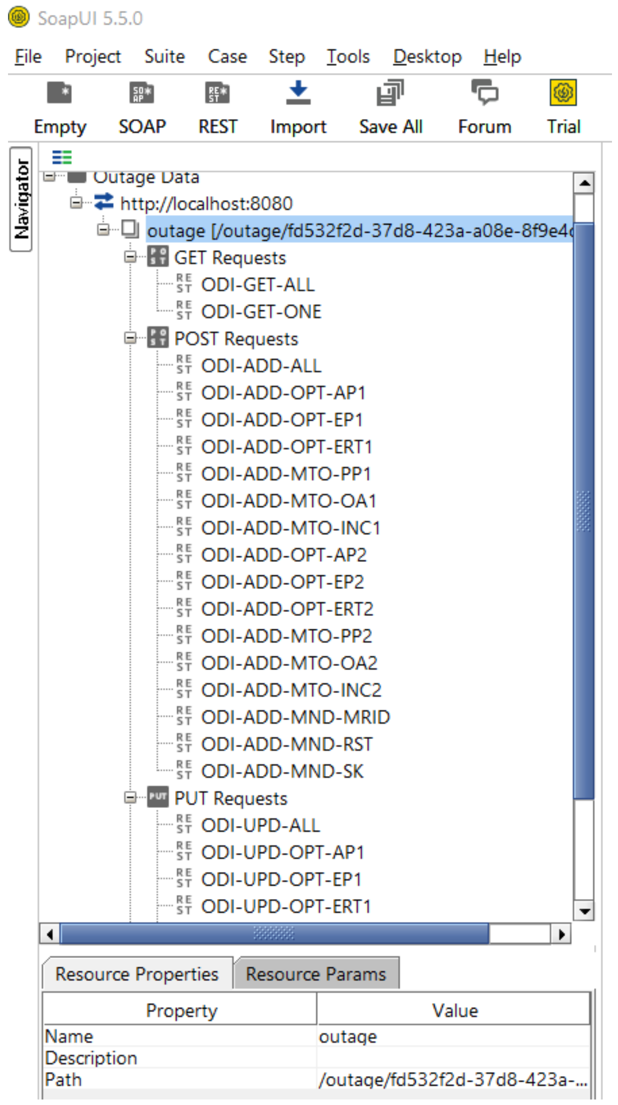
EPRI ELECTRIC POWER RESEARCH INSTITUTE

Test Script Example Use Case – GET/DELETE

- GET messages have no sample XML files due to the nature of how a GET message works
- GET message SoapUI files contain examples for getting all outage data and getting data for a single outage
- DELETE messages follow a similar concept and as a result have no accompanying sample XML files

SoapUI

- SoapUI is a free, open-source tool that allows testing of SOAP or REST web services
- Each use case is prepopulated and only needs the proper URI changed
- The file is not immutable and additional requests can be saved to it as needed

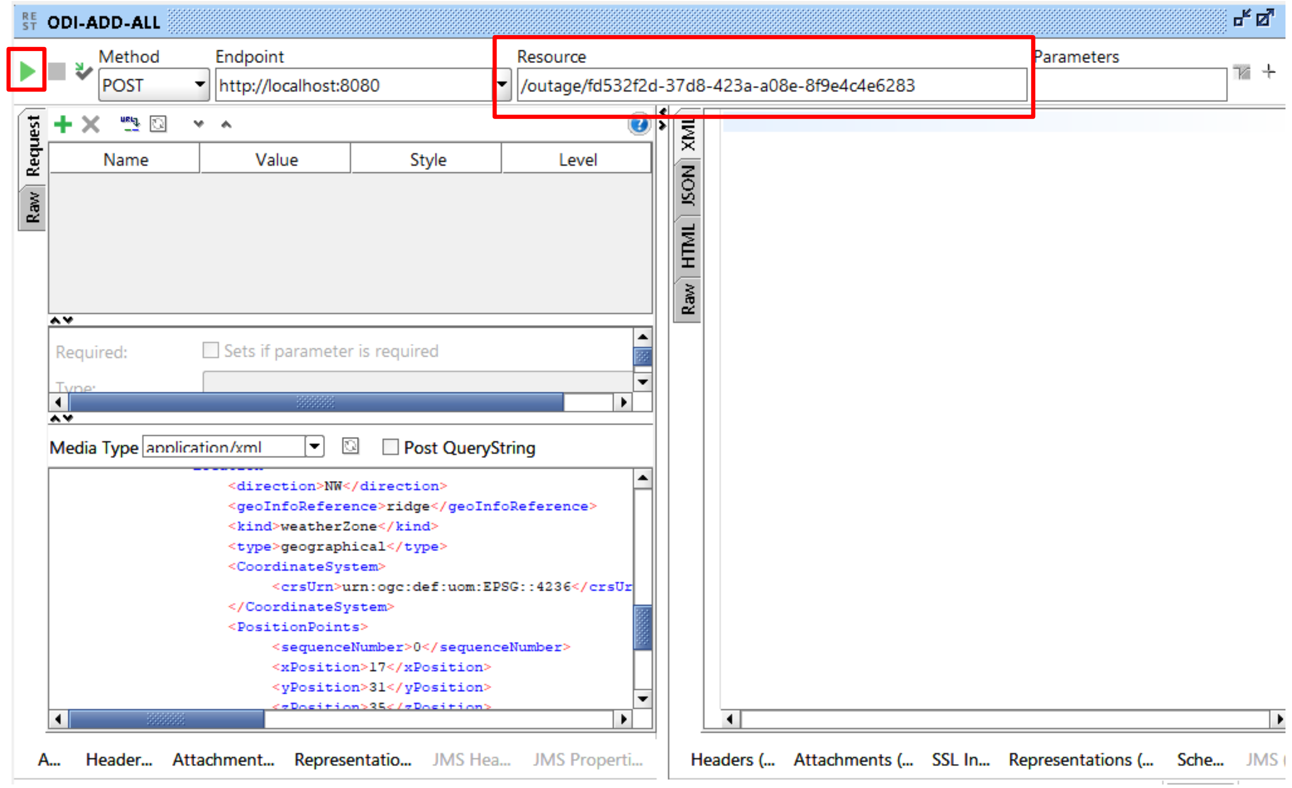


```
Outage Data
  http://localhost:8080
    outage [/outage/fd532f2d-37d8-423a-a08e-8f9e4
      GET Requests
        ODI-GET-ALL
        ODI-GET-ONE
      POST Requests
        ODI-ADD-ALL
        ODI-ADD-OPT-AP1
        ODI-ADD-OPT-EP1
        ODI-ADD-OPT-ERT1
        ODI-ADD-MTO-PP1
        ODI-ADD-MTO-OA1
        ODI-ADD-MTO-INC1
        ODI-ADD-OPT-AP2
        ODI-ADD-OPT-EP2
        ODI-ADD-OPT-ERT2
        ODI-ADD-MTO-PP2
        ODI-ADD-MTO-OA2
        ODI-ADD-MTO-INC2
        ODI-ADD-MND-MRID
        ODI-ADD-MND-RST
        ODI-ADD-MND-SK
      PUT Requests
        ODI-UPD-ALL
        ODI-UPD-OPT-AP1
        ODI-UPD-OPT-EP1
        ODI-UPD-OPT-ERT1
```

Property	Value
Name	outage
Description	
Path	/outage/fd532f2d-37d8-423a-...

SoapUI – Getting Started

- Change “Resource” to point to web service URI
- When ready, click the “Play” button to send the message
- Tool can be used query real outage data as SoapUI provides the ability to act as a real web service client



ODI-ADD-ALL

Method: POST
Endpoint: http://localhost:8080
Resource: /outage/fd532f2d-37d8-423a-a08e-8f9e4c4e6283
Parameters

Name	Value	Style	Level

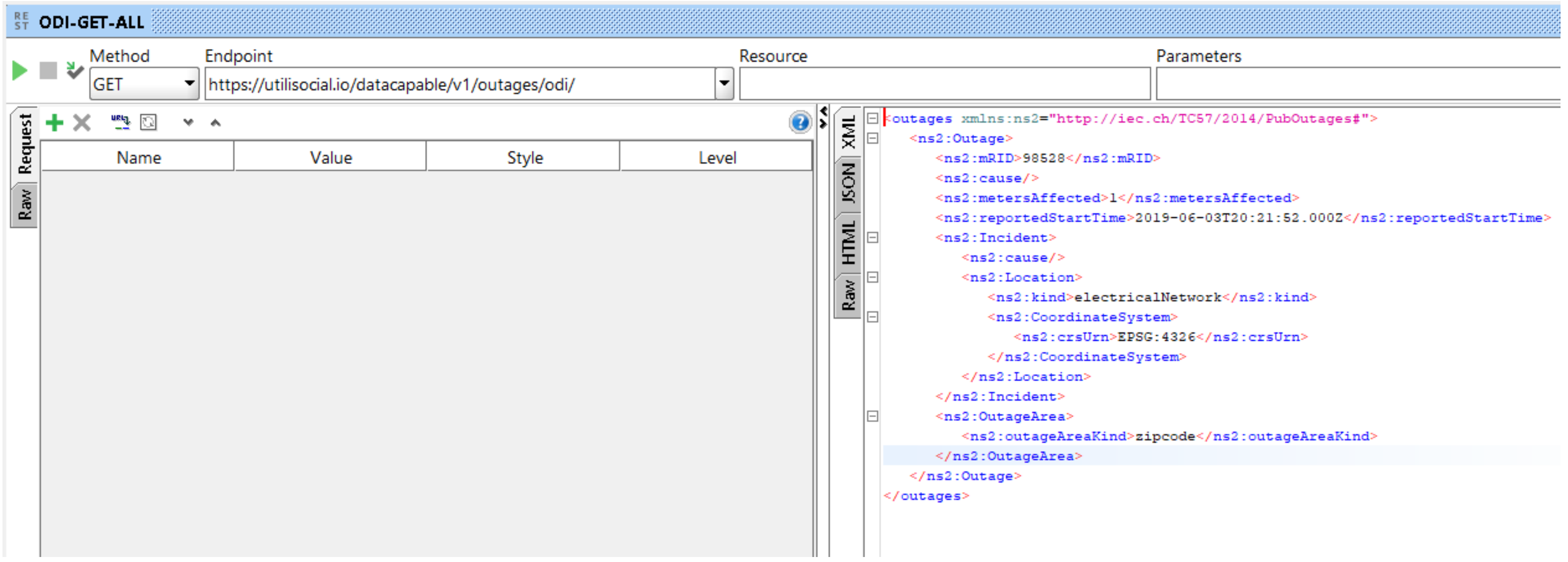
Required: Sets if parameter is required

Media Type: application/xml
Post QueryString

```
        <direction>NW</direction>
        <geoInfoReference>ridge</geoInfoReference>
        <kind>weatherZone</kind>
        <type>geographical</type>
        <CoordinateSystem>
            <crsUrn>urn:ogc:def:uom:EPSG::4236</crsUr
        </CoordinateSystem>
        <PositionPoints>
            <sequenceNumber>0</sequenceNumber>
            <xPosition>17</xPosition>
            <yPosition>31</yPosition>
            <zPosition>35</zPosition>
```

SoapUI – Getting Started

- Alternatively, everything can be put under “Endpoint”
- More information on how to use SoapUI for this test script can be found in the introduction



ODI-GET-ALL

Method: GET

Endpoint: https://utilisocial.io/datacapable/v1/outages/odi/

Resource:

Parameters:

```xml
<outages xmlns:ns2="http://iec.ch/TC57/2014/PubOutages#">
  <ns2:Outage>
    <ns2:mRID>98528</ns2:mRID>
    <ns2:cause/>
    <ns2:metersAffected>1</ns2:metersAffected>
    <ns2:reportedStartTime>2019-06-03T20:21:52.000Z</ns2:reportedStartTime>
    <ns2:Incident>
      <ns2:cause/>
      <ns2:Location>
        <ns2:kind>electricalNetwork</ns2:kind>
        <ns2:CoordinateSystem>
          <ns2:crsUrn>EPSG:4326</ns2:crsUrn>
        </ns2:CoordinateSystem>
      </ns2:Location>
    </ns2:Incident>
    <ns2:OutageArea>
      <ns2:outageAreaKind>zipcode</ns2:outageAreaKind>
    </ns2:OutageArea>
  </ns2:Outage>
</outages>
```

- More information about SoapUI in general can be found at https://www.soapui.org/

EPRI’s CIM Test Harness

- CIM Test Harness currently supports:
  - IEC 61968-5
  - IEC 61968-6
  - IEC 61968-9
- Plans for the near future include adding:
  - IEC 61968-8
  - ODI (both SOAP and RESTful implementations)
- Utilities wishing to test their own services can use the Test Harness for message validation

Questions?

Together…Shaping the Future of Electricity