# XML-to-JSON-Schema README v0.5

George Moyano

June 2017

## Introduction

XML-to-JSON-Schema generates JSON Schema from XML Schema.

The process is de-composed as follows:

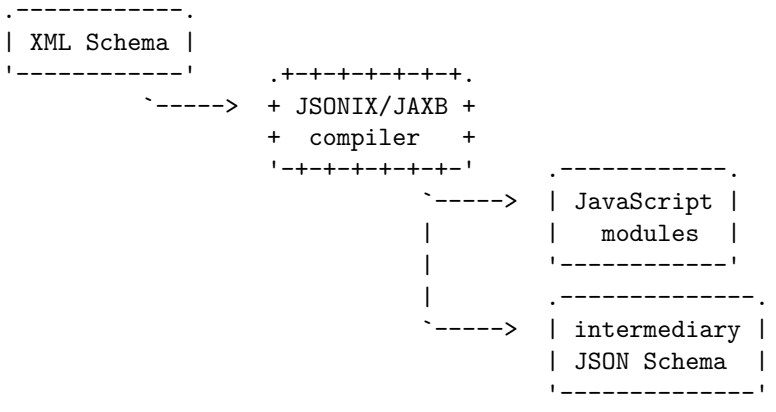
1. Convert XML Schema
2. Extract XML Schema Content
3. Generate JSON Schema Components
4. De-Reference JSON Schema Components

Each automated task is discretely implemented for increased maintainability, modularity, and portability. A project folder for each task contains a README.md file with more details, to include, configuration and usage.

The remainder of this document provides an overview description of each task in the pipeline for generating JSON Schema from XML Schema.

### Convert XML Schema

This task takes each data component described in XML Schema and re-describes it in JavaScript modules and JSON schema.



The JSON schema produced by this task is considered intermediary for purposes of schema-language binding and data modeling. In many cases, further processing is required for programming language schema binders. For example, the implementation of JSON constructs, such as “anyof” and “allOf”, are not well-supported. In addition, the decomposition of general and complex JSON schema into specialized and discrete JSON schema increases composability and re-useability.

#### Software Components

| Library | Language | Package |
|---|---|---|
| JSONIX 2.4.1 | JavaScript | NPM |
| jsonix-schema-compiler 2.3.9 | JavaScript | NPM |
| JAXB | Java | java-1.8.0-openjdk |

| Binary | Package |
|---|---|
| Node v6.9.1 | nodejs |
| XJC 2.2.8 | java-1.8.0-openjdk |

## Extract XML Schema Content

This task extracts valuable XML Schema content ignored by Java XJC/JAXB, see Add support for XML schema facets and documentation #917.

XML documentation and facets for each data component are converted and formatted as JSON data structures. Facets include value constraints such as numeric maximum and minimum, and character string length.

```
.------------.
| XML Schema |
'------------'     .+-+-+-+-+-+-+.
          `----->  + XML content +
                   +  extractor  +
                   '-+-+-+-+-+-+-'     .-----------------.
                             `------>  | data component |
                             |         | documentation  |
                             |         '-----------------'
                             |         .-----------------.
                             `------>  | data component |
                                       |     facets     |
                                       '-----------------'
```

#### Software Components

| Library | Language | Package |
|---|---|---|
| extract-xml-schema-documentation.xsl | XSL | n/a |

| Library | Language | Package |
|---|---|---|
| extract-xml-schema-facets.xsl | XSL | n/a |
| SaxonEE9-7-0-18J* | Java | Saxonica |

*SaxonEE is the purchased product, however, most XSL 2.0-compatible XML transformers should meet expectations.

| Binary | Package |
|---|---|
| Node v6.9.1 | nodejs |
| GNU bash v4.2.46 | x86_64-redhat-linux-gnu |

## Generate JSON Schema Components

This task generates JSON Schema components from the outputs of the Convert and Extract tasks.

```
.------------.
| JavaScript |
|   modules  |`-------->.
'------------'          |
.--------------.        |
| intermediary |        |
| JSON Schema  |`------->:
'--------------'        |       .+-+-+-+-+-+.
.----------------.      :----->  +  schema   +
| data component |      |       + generator +
| documentation  |`----->:       '+-+-+-+-+-+'     .-------------.
'----------------'      |                `----->  | JSON schema |
.----------------.      |                         | components  |
| data component |      |                         '-------------'
|     facets     |`---->'
'----------------'
```

### JSON Schema References

Since this task decomposes general and complex JSON schema into specialized and discrete JSON schema, schema components may reference other JSON schema components to complete a conceptual representation.

For example, the following NIEM Crash schema snippet references ('$ref') the identifier for the JSON schema that describes the property "Location2DGeospatialCoordinate":

```json
{
  "title": "Crash",
  "id": "http://release.niem.gov/niem/domains/jxdm/5.1/#Crash",
  "type": "object",
```

```
  "properties": {
    "Location": {
      "id": "http://release.niem.gov/niem/niem-core/3.0/#Location",
      "type": "object",
      "properties": {
        "Location2DGeospatialCoordinate": {
            $ref": "http://release.niem.gov/niem/niem-core/3.0
              /#Location2DGeospatialCoordinate"
        }
      }
    }
  }
}
```

**Software Components**

| Library | Language | Package |
|---|---|---|
| schema-generator.js | JavaScript | n/a |

| Binary | Package |
|---|---|
| Node v6.9.1 | nodejs |

## De-Reference JSON Schema Components

This task creates a JSON Schema that is de-referenced. In other words, references ('$ref') are replaced with the contents of JSON schema components.

For example, a snippet of the de-referenced Crash component is depicted below:

```JSON
{
  "title": "Crash",
  "id": "http://release.niem.gov/niem/domains/jxdm/5.1/#Crash",
  "type": "object",
  "properties": {
    "Location": {
      "id": "http://release.niem.gov/niem/niem-core/3.0/#Location",
      "type": "object",
      "properties": {
        "Location2DGeospatialCoordinate": {
          "id": "http://release.niem.gov/niem/niem-core/3.0
            /#Location2DGeospatialCoordinate",
          "type": "object",
          "properties": {
            "GeographicCoordinateLatitude": {
              "id": "http://release.niem.gov/niem/niem-core/3.0
                /#GeographicCoordinateLatitude",
```

```
            "type": "object",
            "properties": {
              "LatitudeDegreeValue": {
                "id": "http://release.niem.gov/niem/niem-core/3.0
                  /LatitudeDegreeValue",
                "type": "object",
                "properties": {
                  "value": {
                    "type": "number"
                  }
                }
              }
            }
          },
          "GeographicCoordinateLongitude": {
            "id": "http://release.niem.gov/niem/niem-core/3.0
              /#GeographicCoordinateLongitude",
            "type": "object",
            "properties": {
              "LongitudeDegreeValue": {
                "id": "http://release.niem.gov/niem/niem-core/3.0
                  /LongitudeDegreeValue",
                "type": "object",
                "properties": {
                  "value": {
                    "type": "number"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

The resulting de-referenced JSON schema may be used to validate JSON instances. To validate instances, use a JSON Schema validator with the de-referenced JSON schema.

### Software Components

| Library | Language | Package |
|---|---|---|
| json-schema-ref-parser v3.1.2 | JavaScript | NPM |

| Binary | Package |
|---|---|
| Node v6.9.1 | nodejs |

## Conclusion

XML-to-JSON-Schema generates JSON Schema from XML Schema with discrete automated tasks:

1. Convert XML Schema
2. Extract XML Schema Content
3. Generate JSON Schema Components
4. De-Reference JSON Schema Components

Each automated task has a specific project folder with a README.md file that includes configuration and usage. Their discrete implementation increases maintainability, modularity, and portability.

## Appendix A - Use Case: NIEM XML Schema Components

### Search NIEM XML Schema Components

### Generate NIEM Schema Subset

### Configure XML-to-JSON-Schema-Converter tasks