

odmlib: A Python library for creating ODM and Define-XML

Sam Hume

PHUSE US Connect 2022

OS01

04 May 2022



Agenda

- Getting started
- Models
- Serialization
- Creating Objects
- Generating ODM Documents
- Validating ODM Documents
- Processing ODM Documents
- Conclusion

Getting Started

What you need to know about odmlib to get started...

What is odmlib?



A Python package that simplifies working with ODM and Define-XML



Provides an object-oriented interface



Supports serializations in XML, JSON, and Python dictionaries



Models for ODM, Define-XML, Controlled Terminology, and Dataset-XML



Simplifies creating proprietary ODM or Define-XML extensions



Supports conformance checks and schema validation

Useful odmlib details

- Open-source using the MIT license
- Comes with a number of useful and instructive example programs
- Available on GitHub
- Can be installed using pip
- Part of COSA
- Under development

Models

The models that are included in odmlib and how to create your own

Models Included in odmlib

Model Name	Description
odm_1_3_2	Model for implementing ODM v1.3.2
define_2_1	Model for implementing Define-XML v2.1
define_2_0	Model for implementing Define-XML v2.0
ct_1_1_1	Model for implement CT-XML v1.1.1 (controlled terminology)
dataset_1_0_1	Model for implementing Dataset-XML v1.0.1
library_define_1_0 library_odm_1_0	Models for retrieving Library-XML from the CDISC Library using the application/odm media type are available in odmlib_examples

Understanding the ODM v1.3.2 Model

```
class ODM(OE.ODMElement):
    Description = T.String(required=False)
    FileType = T.ValueSetString(required=True)
    Granularity = T.ValueSetString(required=False)
    Archival = T.ValueSetString(required=False)
    FileOID = T.OID(required=True)
    CreationDateTime = T.DateTimeString(required=True)
    PriorFileOID = T.OIDRef(required=False)
    AsOfDateTime = T.DateTimeString(required=False)
    ODMVersion = T.ValueSetString(required=False)
    Originator = T.String(required=False)
    SourceSystem = T.String(required=False)
    SourceSystemVersion = T.String(required=False)
    schemaLocation = T.String(required=False, namespace="xs")
    ID = T.ID()
    Study = T.ODMListObject(element_class=Study)
    AdminData = T.ODMListObject(element_class=AdminData)
    ReferenceData = T.ODMListObject(element_class=ReferenceData)
    ClinicalData = T.ODMListObject(element_class=ClinicalData)
    Association = T.ODMListObject(element_class=Association)
```

ODM v1.3.2 odmlib model: ODM class

3.1 ODM

Body:

([Study](#)*, [AdminData](#)*, [ReferenceData](#)*, [ClinicalData](#)*, [Association](#)*, [ds:Signature](#)*)

Attributes:

Description	text	(optional)
FileType	(Snapshot Transactional)	
Granularity	(All Metadata AdminData ReferenceData AllClinicalData SingleSite SingleSubject)	(optional)
Archival	(Yes No)	(optional)
FileOID	oid	
CreationDateTime	datetime	
PriorFileOID	oidref	(optional)
AsOfDateTime	datetime	(optional)
ODMVersion	(1.2 1.2.1 1.3 1.3.1 1.3.2)	(optional)
Originator	text	(optional)
SourceSystem	text	(optional)
SourceSystemVersion	text	(optional)
ID	ID	(optional)

ODM v1.3.2 specification: ODM element

```
class ItemGroupDef(ODM.ItemGroupDef):
    OID = ODM.ItemGroupDef.OID
    Name = ODM.ItemGroupDef.Name
    Repeating = ODM.ItemGroupDef.Repeating
    IsReferenceData = ODM.ItemGroupDef.IsReferenceData
    SASDatasetName = ODM.ItemGroupDef.SASDatasetName
    Domain = ODM.ItemGroupDef.Domain
    Purpose = ODM.ItemGroupDef.Purpose
    Structure = T.String(required=True, namespace="def")
    ArchiveLocationID = T.String(namespace="def")
    CommentOID = T.OIDRef(namespace="def")
    IsNonStandard = T.ValueSetString(namespace="def")
    StandardOID = T.OIDRef(namespace="def")
    HasNoData = T.ValueSetString(namespace="def")
    Description = ODM.ItemGroupDef.Description
    ItemRef = ODM.ItemGroupDef.ItemRef
    Alias = ODM.ItemGroupDef.Alias
    Class = T.ODMObject(element_class=Class, namespace="def")
    leaf = T.ODMObject(required=True, element_class=leaf, namespace="def")
```

Extending models in odmlib

Serialization

Serialize odmlib objects as XML, JSON, or as a Python dictionary

Define-XML v2.1 Serialization Examples

```
<MetaDataVersion OID="MDV.COSA.IGD.001" Name="ItemGroupDefDemo001"
  Description="ItemGroupDef COSA Demo" def:DefineVersion="2.1.0">
  <def:Standards>
    <def:Standard OID="STD.1" Name="SDTMIG" Type="IG" Version="3.2" Status="Final"/>
    <def:Standard OID="STD.2" Name="CDISC/NCI" Type="CT" PublishingSet="SDTM"
      Version="2021-12-17" Status="Final"/>
  </def:Standards>
  <ItemGroupDef OID="IG.VS" Name="VS" Repeating="Yes" Domain="VS" SASDatasetName="VS"
    IsReferenceData="No" Purpose="Tabulation" def:ArchiveLocationID="LF.VS"
    def:Structure="One record per vital sign measurement per visit per subject"
    def:StandardOID="STD.1" def:IsNonStandard="Yes" def:HasNoData="Yes">
    <Description>
      <TranslatedText xml:lang="en">Vital Signs</TranslatedText>
    </Description>
    <ItemRef ItemOID="IT.STUDYID" Mandatory="Yes" OrderNumber="1" KeySequence="1"/>
    <ItemRef ItemOID="IT.VS.DOMIAN" Mandatory="Yes" OrderNumber="2"/>
    <ItemRef ItemOID="IT.USUBJID" Mandatory="Yes" OrderNumber="3" KeySequence="2"/>
    <ItemRef ItemOID="IT.VS.VSSEQ" Mandatory="Yes" OrderNumber="4"/>
    <ItemRef ItemOID="IT.VS.VSTESTCD" Mandatory="Yes" OrderNumber="5" KeySequence="3"/>
    <ItemRef ItemOID="IT.VS.VSTEST" Mandatory="Yes" OrderNumber="6"/>
    <def:Class Name="FINDINGS"/>
  </ItemGroupDef>
</MetaDataVersion>
```

XML Serialization

```
"MetaDataVersion": {
  "OID": "MDV.COSA.IGD.001",
  "Name": "ItemGroupDefDemo001",
  "Description": "ItemGroupDef COSA Demo",
  "DefineVersion": "2.1.0",
  "Standards": {
    "Standard": [
      {
        "OID": "STD.1",
        "Name": "SDTMIG",
        "Type": "IG",
        "Version": "3.2",
        "Status": "Final"
      },
      {
        "OID": "STD.2",
        "Name": "CDISC/NCI",
        "Type": "CT",
        "PublishingSet": "SDTM",
        "Version": "2021-12-17",
        "Status": "Final"
      }
    ]
  },
  "ItemGroupDef": [
    {
      "OID": "IG.VS",
```

JSON Serialization

Creating Objects

Creating your first odmlib objects using the Define-XML v2.1 model

```
import odmlib.define_2_1.model as DEFINE
odm = DEFINE.ODM(FileOID="DEF.COSA.DEMO",
                  AsOfDateTime=current_datetime,
                  CreationDateTime=current_datetime,
                  ODMVersion="1.3.2",
                  FileType="Snapshot",
                  Originator="Sam Hume",
                  SourceSystem="odmlib",
                  SourceSystemVersion="0.1.4",
                  Context="Other")
```

Create an ODM Object

```
study = DEFINE.Study(OID="ST.DEFINE.COSA.001")
study.GlobalVariables.StudyName = DEFINE.StudyName(_content="TEST Define-XML ItemGroupDef")
study.GlobalVariables.StudyDescription = DEFINE.StudyDescription(_content="ItemGroupDef 001")
study.GlobalVariables.ProtocolName = DEFINE.ProtocolName(_content="Define-XML ItemGroupDef")
odm.Study = study
```

Create a Study Object

```
mdv = DEFINE.MetadataVersion(OID="MDV.COSA.IGD.001", Name="ItemGroupDefDemo001",  
                             Description="ItemGroupDef COSA Demo", DefineVersion="2.1.0")
```

Create a MetadataVersion Object

```
igd = DEFINE.ItemGroupDef(OID="IG.VS",  
                           Name="VS",  
                           Repeating="Yes",  
                           Domain="VS",  
                           SASDatasetName="VS",  
                           IsReferenceData="No",  
                           Purpose="Tabulation",  
                           ArchiveLocationID="LF.VS",  
                           Structure="One record per vital sign measurement per visit per subject",  
                           StandardOID="STD.1",  
                           IsNonStandard="Yes",  
                           HasNoData="Yes")
```

Create an ItemGroupDef Object

```
igd.Description.TranslatedText.append(DEFINE.TranslatedText(_content="Vital Signs", lang="en"))
igd.ItemRef.append(DEFINE.ItemRef(ItemOID="IT.STUDYID", Mandatory="Yes", OrderNumber=1,
                                   KeySequence=1))
igd.ItemRef.append(DEFINE.ItemRef(ItemOID="IT.VS.DOMIAN", Mandatory="Yes", OrderNumber=2))
igd.ItemRef.append(DEFINE.ItemRef(ItemOID="IT.USUBJID", Mandatory="Yes", OrderNumber=3,
                                   KeySequence=2))
igd.ItemRef.append(DEFINE.ItemRef(ItemOID="IT.VS.VSSEQ", Mandatory="Yes", OrderNumber=4))
igd.ItemRef.append(DEFINE.ItemRef(ItemOID="IT.VS.VSTESTCD", Mandatory="Yes", OrderNumber=5,
                                   KeySequence=3))
igd.ItemRef.append(DEFINE.ItemRef(ItemOID="IT.VS.VSTEST", Mandatory="Yes", OrderNumber=6))
```

Create ItemRef Objects

```
try:
    ir = DEFINE.ItemRef(Mandatory="Yes", OrderNumber=1)
except ValueError as ve:
    print(f"Error creating ItemRef: {ve}")
```

Validation During Object Creation

```
odm.Study.MetaDataVersion = mdv  
odm.Study.MetaDataVersion.ItemGroupDef.append(igd)
```

Build the Object Hierarchy

Generating ODM Documents

Using the obmlib objects, generate a Define-XML v2.1 file in XML and JSON

```
odm.write_xml(odm_file="./data/cosa_define_demo.xml")  
odm.write_json(odm_file="./data/cosa_define_demo.json")
```

Generate Define-XML Files

Validating ODM Documents

Schema validate the newly generated Define-XML v2.1 file

```
validator = P.ODMSchemaValidator(schema_file)
try:
    # update the path to reflect your system
    validator.validate_file("./data/cosa_define_demo.xml")
    print("define-XML schema validation completed successfully...")
except P.OdmLibSchemaValidationError as ve:
    print(f"schema validation errors: {ve}")
```

Validate the Define-XML File

Processing ODM Documents

Load and process the newly created Define-XML v2.1 file

```
from odmlib import define_loader as DL, loader as LD
loader = LD.ODMLoader(DL.XMLDefineLoader(model_package="define_2_1",
                                         ns_uri="http://www.cdisc.org/ns/def/v2.1"))
loader.open_odm_document("./data/cosa_define_demo.xml")
```

Open the Define-XML File

```
odm = loader.load_odm()  
print(f"Study OID is {odm.Study.OID}")  
print(f"Study Name is {odm.Study.GlobalVariables.StudyName}")  
print(f"Study Description is {odm.Study.GlobalVariables.StudyDescription}")  
print(f"Protocol Name is {odm.Study.GlobalVariables.ProtocolName}")
```

Load and Process the Define-XML File

Conclusion

Where to find the code and more information

Code for the Example Covered Today

- Available as a Jupyter Notebook

https://github.com/swhume/odmlib_examples/tree/master/notebooks

- Available as a code snippet in Gist

<https://gist.github.com/c686b40a6356c48f42b3bd2bef53c980>

- Generate a more complete Define-XML v2.1

https://github.com/swhume/odmlib_examples/tree/master/xlsx2define2-1

- Conference paper includes more detail and code: PAP_OS01

odmli Next Steps



Additional testing



Complete the documentation



DevOps CI/CD



More example programs

References

- COSA Repository Directory
<https://cosa.cdisc.org/>
- COSA LinkedIn Group
<https://www.linkedin.com/groups/12563665/>
- odmlib repository:
<https://github.com/swhume/odmlib>
- odmlib examples repository:
https://github.com/swhume/odmlib_examples
- odmlib documentation:
<https://swhume.github.io/odmlib/>
- blog posts:
<https://swhume.github.io/blog-home.html>
- LinkedIn:
<https://www.linkedin.com/in/sam-hume-dsc>
- CDISC email:
shume@cdisc.org

Thank you!