

Extracting Data Standards Metadata and Controlled Terminology from the CDISC Library using SAS[®] with PROC LUA

Lex Jansen
SAS Institute Inc., USA

PharmaSUG 2021
Paper AD-168

Extracting Data Standards Metadata and Controlled Terminology from the CDISC Library using SAS® with PROC LUA

Biography



Lex is a Principal Solution Consultant at SAS Institute and has more than 25 years of experience in designing and implementing solutions in Life Sciences.

- One of the main developers of the CDISC Define-XML v2.x standard
- Developer of the CDISC / PhUSE Define-XML v2.x stylesheet
- One of the main developers of the CDISC Analysis Results Metadata v1.0 for Define-XML v2.x extension
- Owner / developer of the SAS Proceedings Library at <https://www.lexjansen.com> with links to almost 35,000 SAS papers

Agenda

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

The CDISC Library

Introduction

- Single, trusted, authoritative source of CDISC Data Standards metadata and Controlled Terminology
- Cloud-based metadata repository (MDR) on the Microsoft Azure platform
- Uses linked data and a REST API to deliver CDISC Data Standards metadata and Controlled Terminology in a machine-readable format to software applications that automate standards-based processes
- CDISC Library provides access to new relationships between standards
- Available to CDISC Members and Open Source Developers
- Need a separate CDISC Library account, which is different from the CDISC account to access the Members Only Area on the CDISC website

The CDISC Library

Common use cases

Retrieve the latest versions of the standards and CT for loading into internal metadata repositories

Build Standards-aware study design tools

Check conformance against standards metadata

Search across all versions of all standards and CT

ETL designer tools

Impact / change management

Novel metadata simplifies task automation (e.g. QRS)

Standards visualization, browsing, and retrieval (e.g. Data Standards Browser)

Retrieve using multiple media-types including ODM and Define-XML

*) Webinar CDISC Library April 2021: Ideas for using the CDISC Library and a Look at What's Coming Next (Anthony Chow, Sam Hume)
<https://www.cdisc.org/events/webinar/cdisc-library-ideas-using-cdisc-library-and-look-whats-coming-next>

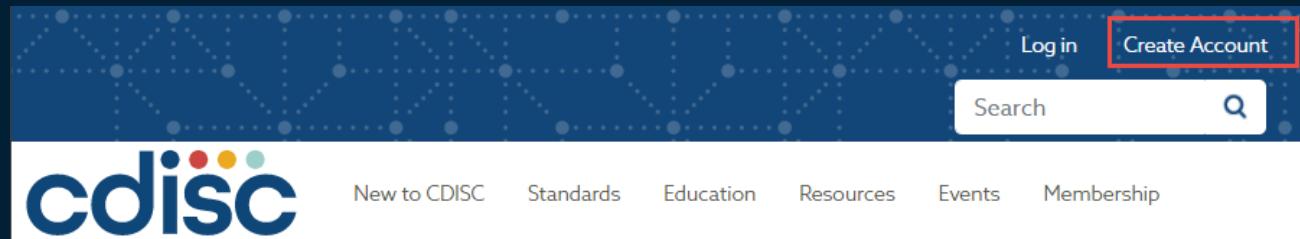
The CDISC Library

Accessing

- Different ways of accessing the CDISC Library:
 - Manual downloads from the CDISC Library Archives
 - Data Standards Browser – including downloads
 - Programmatically through the CDISC Library REST API

All of these require a CDISC account which can be created at cdisc.org

The last 2 – for Data Standards Browser and REST API - also need the CDISC Library account.



The CDISC Library

Accessing

[Log out](#)[My Account](#)[New to CDISC](#)[Standards](#)[Education](#)[Resources](#)[Events](#)[Membership](#)[Members Only](#)[Home](#) / [CDISC Library](#)[Overview](#)[Available Content](#)[Request an Account](#)[FAQs](#)[Benefits Summary](#)[Up](#)

Member Benefits

[Industry Job Board Posting](#)[Interchange Presentations](#)[Online Training Credit](#)[Webinar Archive](#)

Members Only

[CDISC Library](#)[CDISC Library Archives](#)

Thanks for being a CDISC Platinum Member!

Access to the CDISC Library is a benefit of CDISC Platinum Membership. Please refer to the [Benefits Sum](#)

Please ensure this information is communicated throughout your organization and carefully consider who should be allowed access to this important tool.

The request form will require your organization to enter contact information for an [Account Administrator](#), [Account User](#), and [Technical Contact](#).

The CDISC Library

CDISC Library Archives

- Formerly known as CDISC SHARE Exports
- Downloads of:
 - Machine-readable metadata published from CDISC Library. (Excel, CSV, ODM XML, Define-XML, RDF)
 - PDF documents (a standard's specifications, implementation guides (IG), or user guides (UG))
 - Diff (Difference) files that summarize changes between standards
- <https://www.cdisc.org/members-only/cdisc-library-archives>

Members Only

CDISC Library

CDISC Library Archives



The CDISC Library

CDISC Library Archives

SDTM

Date Posted to Archive	Content	Version	Type	Download Files
SDTMIG 3.3				
2018-11-19	SDTM IG	3.3	Metadata	Excel
2018-11-19	SDTM Diff	3.3 - 3.2	Metadata	Excel
2020-06-26	SDTM IG	3.3	Document	PDF
SDTM 1.7				
2018-11-19	SDTM	1.7	Metadata	Excel
2018-11-19	SDTM Diff	1.7 - 1.6	Metadata	Excel
SDTM Terminology				
2021-01-21	SDTM Terminology	2020-12	Metadata	ODM v1.3.1 RDF Excel Diff
2020-11-25	SDTM Terminology	2020-11	Metadata	ODM v1.3.1 RDF Excel Diff

The CDISC Library

Data Standard Browser

The screenshot shows the CDISC Library Data Standards Browser. On the left, a sidebar menu lists categories: Dashboard, Data Collection, Data Tabulation, Data Analysis, Terminology, Draft Content, ADaMIG v1.3, and SDTMIG v3.4. Red arrows point from the 'Data Collection', 'Data Tabulation', 'Data Analysis', and 'Terminology' menu items to their respective boxes in the main content area. The main content area displays four boxes: CDASH, SDTM, SEND, and ADaM under 'Data Collection'; SDTM and SEND under 'Data Tabulation'; ADaM under 'Data Analysis'; and Controlled Terminology under 'Terminology'. A URL at the bottom of the main content area is <https://library.cdisc.org/browser>.

cdisc LIBRARY

Data Standards Browser

- Dashboard
- Expand All
- ▼ Data Collection
- ▼ Data Tabulation
- ▼ Data Analysis
- ▼ Terminology
- ▲ Draft Content
- ADaMIG v1.3 ⚠
- SDTMIG v3.4 ⚠

CDASH

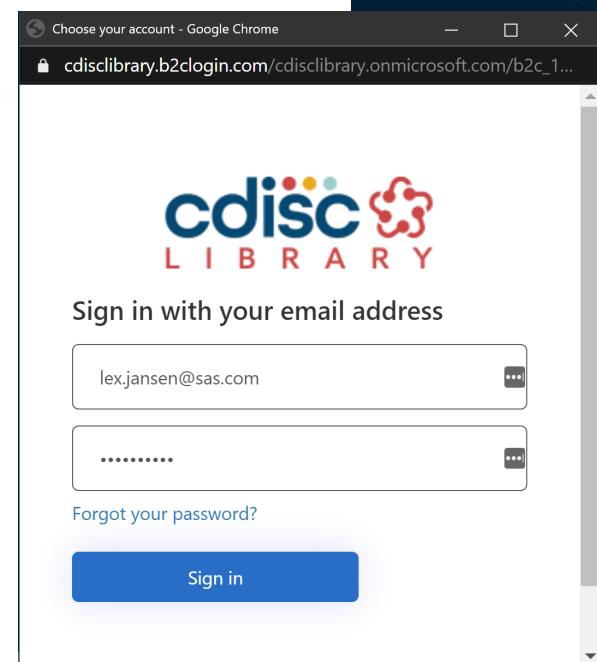
SDTM

SEND

ADaM

Controlled Terminology

<https://library.cdisc.org/browser>



The CDISC Library

Data Standard Browser

cdisc LIBRARY Data Standards Browser

Search

Dashboard

Expand All

Data Collection

Data Tabulation

- SDTM v1.8
- SDTM v1.7
- SDTM v1.6
- SDTM v1.5
- SDTM v1.4
- SDTM v1.3
- SDTM v1.2
- SDTMIG-MD v1.1
- SDTMIG v3.3**
- SDTMIG-PGx v1.0
- SDTMIG-AP v1.0
- SDTMIG v3.2
- SDTMIG-MD v1.0
- SDTMIG v3.1.3
- SDTMIG v3.1.2
- SENDIG v3.1.1
- SENDIG-AR v1.0
- SENDIG-DART v1.1
- SENDIG v3.1
- SENDIG v3.0

Classes

- General Observations
- Special-Purpose**
- Interventions
- Events
- Findings
- Findings About
- Trial D

Study Reference

Data Sets

- CO
- DM**
- SE
- SM
- SV

Special-Purpose DM

Name Structure

Demographics One record per subject

Description

A special-purpose domain that includes a set of essential standard variables that describe each subject in a clinical study. It is the parent domain for all CDISC Controlled Terminology, DOMAIN, C49572, 2018-06-29)

Demographics

Ordinal ↑	Name	Label	Description	Data Type	Role	Core
1	STUDYID	Study Identifier	Unique identifier for a study.	Char	Identifier	Req
2	DOMAIN	Domain Abbreviation	Two-character abbreviation for the domain.	Char	Identifier	Req
3	USUBJID	Unique Subject Identifier	Identifier used to uniquely identify a subject across all studies for all applications or	Char	Identifier	Req

Details

Ordinal
3

Name
USUBJID

Label
Unique Subject Identifier

Description
Identifier used to uniquely identify a subject across all studies for all applications or submissions in a product. This must be a unique number, and a compound identifier formed by concatenating SITEID-SUBJID.

Data Type
Char

Role
Identifier

Core
Req

Code List
--

Described Value Domain
--

Implements
USUBJID

The CDISC Library

Data Standard Browser



Data Standards Browser

Search



SDTMIG v3.3

Status Effective Date Implements
Final 2018-11-20 SDTM v1.7

Classes

General Observations Special-Purpose Interventions Events Findings Findings About Trial Design
Relationship Study Reference

Data Sets

CO DM SE SM SV

Special-Purpose

DM

Name Structure

Demographics One record per subject

Description

A special-purpose domain that includes a set of essential standard variables that describe each subject in a clinical study. It is the parent domain for all other observations for human clinical subjects. (Source: CDISC Controlled Terminology, DOMAIN, C49572, 2018-06-29)

Demographics

Export
Comma-Separated Values (CSV)
Microsoft® Excel® (XLSX)

The CDISC Library

Data Standard Browser

 Data Standards Browser

Search ⚙️

[Dashboard](#) [Expand All](#)

[Data Collection](#)

[Data Tabulation](#)

[SDTM v1.8](#)

[SDTM v1.7](#)

[SDTM v1.6](#)

[SDTM v1.5](#)

[SDTM v1.4](#)

[SDTM v1.3](#)

[SDTM v1.2](#)

[SDTMIG-MD v1.1](#)

[SDTMIG v3.3](#)

[SDTMIG-PGx v1.0](#)

[SDTMIG-AP v1.0](#)

[SDTMIG v3.2](#)

[SDTMIG-MD v1.0](#)

[SDTMIG v3.1.3](#)

[SDTMIG v3.1.2](#)

[SENDIG v3.1.1](#)

[SENDIG-AR v1.0](#)

[SENDIG-DART v1.1](#)

[SENDIG v3.1](#)

[SENDIG v3.0](#)

[Data Analysis](#)

[Terminology](#)

Findings LB

Name Laboratory Test Results **Structure** One record per lab test per time point per visit per subject

Description
A findings domain that contains laboratory test data such as hematology, clinical chemistry and urinalysis. This domain does not include microbiology or pharmacokinetic data, which are stored in separate domains. (Source: CDISC Controlled Terminology, DOMAIN, C49592, 2018-06-29)

Laboratory Test Results

Ordinal ↑	Name	Label	Description	Data Type	Role	Core	Code List	Described Value Domain	Impl
8	LBTESTCD	Lab Test or Examination Short Name.	Short name of the measurement, test, or examination described in LBTEST. It can be used as a column name when converting a dataset from a vertical to a horizontal format. The value in LBTESTCD cannot be longer than 8 characters, nor can it start with a number (e.g., "1TEST" is not valid). LBTESTCD cannot contain characters other than letters, numbers, or underscores. Examples: "ALT", "LDH".	Char	Topic	Req	C65047		-TE
9	LBTEST	Lab Test or Examination Name	Verbatim name of the test or examination used to obtain the measurement finding. Note any test normally performed by a clinical laboratory is considered a lab test. The value in LBTEST cannot be longer than 40 characters. Examples: "Alanine Aminotransferase", "Lactate Dehydrogenase".	Char	Synonym Qualifier	Req	C67154		-TE

test ×



The CDISC Library

Data Standard Browser

cdisc LIBRARY Data Standards Browser Search ⚙️

Dashboard < Expand All

Data Collection

Data Tabulation

- SDTM v1.8
- SDTM v1.7
- SDTM v1.6
- SDTM v1.5
- SDTM v1.4
- SDTM v1.3
- SDTM v1.2
- SDTMIG-MD v1.1
- SDTMIG v3.3
- SDTMIG-PGx v1.0
- SDTMIG-AP v1.0
- SDTMIG v3.2
- SDTMIG-MD v1.0
- SDTMIG v3.1.3
- SDTMIG v3.1.2
- SENDIG v3.1.1
- SENDIG-AR v1.0
- SENDIG-DART v1.1
- SENDIG v3.1

Controlled Terminology Package Effective 2021-03-26

Packages

CDASH CT 2021-03-26 Define-XML CT 2021-03-26 Protocol CT 2021-03-26 **SDTM CT 2021-03-26** SEND CT 2021-03-26

C65047

Extensible: Yes
Submission Value: LBTESTCD
Definition: Terminology used for laboratory test codes of the CDISC Study Data Tabulation Model.
NCI Preferred Term: CDISC SDTM Laboratory Test Code Terminology
Synonyms: Laboratory Test Code

Filter results

Term	Submission Value	Synonyms	Definition	NCI Preferred Term
C177957	2MCA	2-Methylcitrate; 2-Methylcitric Acid; MCA; Methylcitrate; Methylcitric Acid	A measurement of the 2-methylcitrate in a biological specimen.	2-Methylcitrate Measurement

The CDISC Library

Data Standard Browser



Data Standards Browser

vstest



Close search results

Search Results

24 results for "vstest"

Narrow search results

Scope

Product

Scope Value

Select

Reset

- CDASHIG v2.0
- CDASHIG v2.1
- SDTMIG v3.1.2
- SDTMIG v3.1.3
- SDTMIG v3.2
- SDTMIG v3.3
- SENDIG v3.0
- SENDIG v3.1
- SENDIG v3.1.1
- CDASHIG v1.1

VSTEST (Data Collection Field)

{**VSTEST**} (See Section 2.2.)

[Clinical Data Acquisition Standards Harmonization \(CDASH\) User Guide V1.1](#)

VSTEST (SDTM Dataset Variable)

The value in **VSTEST** cannot be longer than 40 characters.

[Standard for Exchange of Nonclinical Data Implementation Guide: Nonclinical Studies Version 3.1 \(Final\)](#)

VSTEST (SDTM Dataset Variable)

The value in **VSTEST** cannot be longer than 40 characters.

[Study Data Tabulation Model Implementation Guide: Human Clinical Trials Version 3.2 \(Final\)](#)

VSTEST (SDTM Dataset Variable)

The value in **VSTEST** cannot be longer than 40 characters.

[Standard for Exchange of Nonclinical Data Implementation Guide: Nonclinical Studies Version 3.1.1 \(Final\)](#)

The CDISC Library

CDISC Library REST API

The screenshot shows a web browser window with two tabs: "CDISC Data Standards Browser" and "User: Profile - CDISC Library API Manager". The URL in the address bar is "api.developer.library.cdisc.org/profile", which is highlighted with a red box. The main content area displays the "Account and API Key" section. It includes "Account details" with fields: Email (lex.jansen@sas.com), First name (Lex), Last name (Jansen), and Registration date (11/02/2020). Below this is a "Subscriptions" section with a table showing one entry: Name (Default subscription to CDISC Library API), Product (CDISC Library API), State (Active), and Action (Cancel). The Primary key and Secondary key fields are also shown. A yellow box highlights the registration date and the primary key values. The top navigation bar includes links for "CDISC Library Browser", "Reports", "API Tester", "API Key" (which is highlighted with a yellow box), and "Sign out".

Account and API Key

Account details

Email	lex.jansen@sas.com
First name	Lex
Last name	Jansen
Registration date	11/02/2020

Subscriptions

Subscription details	Product	State	Action
Name Primary key Secondary key	Default subscription to CDISC Library API XXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXX	Rename Show Regenerate Show Regenerate	CDISC Library API Active Cancel

The CDISC Library

CDISC Library REST API

CDISC Library API ▾

No API selected

Try it ▶

Search operations

+ Group by tag

/mdr/sdtm/{version}

Get SDTM product

Study Data Tabulation Model (SDTM)

Request

GET https://api.library.cdisc.org/api/mdr/sdtm/{version}[?expand]

Request parameters

Name	In	Required	Type	Description
version	template	true	string	CDISC Library Product Version
expand	query	false	boolean	If true, entity links will be hydrated and returned within the response (e.g. Fields, Domains, Classes)

The CDISC Library

CDISC Library REST API

/mdr/sdtm/{version}

Parameters

version

1-7

expand

false

Remove

+ Add parameter

Headers

Cache-Control

no-cache

Remove

api-key

d15230994a52459298be68294

Remove

+ Add header

HTTP response

HTTP/1.1 200 OK

content-length: 683181
content-type: application/json
date: Wed, 14 Apr 2021 02:30:16 GMT
vary: Origin
x-referenceid: 5fcde313-df16-4a3c-8398-937e865a5cce

```
{  
    "_links": {  
        "priorVersion": {  
            "href": "/mdr/sdtm/1-6",  
            "title": "Study Data Tabulation Model Version  
1.6",  
            "type": "Foundational Model"  
        },  
        "self": {  
            "href": "/mdr/sdtm/1-7",  
            "title": "Study Data Tabulation Model Version  
1.7",  
            "type": "Foundational Model"  
        }  
    },  
    "classes": [{  
        "_links": {  
            "parentProduct": {  
                "href": "/mdr/sdtm/1-7"  
            }  
        }  
    }]
```

The CDISC Library

REST API - Documentation

Servers

<https://virtserver.swaggerhub.com/CDISC1/cdisc-library/1.3> - SwaggerHub API Auto Mocking ▾

Authorize



Statuses >

Searches >

default ▾

GET

/mdr/products /mdr/products



GET

/mdr/about /mdr/about



GET

/mdr/lastupdated /mdr/lastupdated



GET

/mdr/products/DataTabulation /mdr/products/DataTabulation



The CDISC Library

REST API - Documentation

200

OK

Media type

application/json

Controls `Accept` header.

Example Value | Schema

Response in JSON

```
{  
  "_links": {  
    "self": {  
      "href": "/mdr/products",  
      "title": "CDISC Library Product List",  
      "type": "CDISC Library Product List"  
    },  
    "data-collection": {  
      "_links": {  
        "self": {  
          "href": "/mdr/products/DataTabulation",  
          "title": "Product Group Data Tabulation",  
          "type": "CDISC Library Product Group"  
        },  
        "cdash": [  
          {  
            "href": "/mdr/cdash/1-1",  
            "title": "Clinical Data Acquisition Standards Harmonization Model Version 1.1",  
            "type": "Foundational Model"  
          }  
        ]  
      }  
    }  
  }  
}
```

The CDISC Library

REST API - Documentation

200
OK

Media type

application/xml

Controls `Accept` header.

Example Value | Schema

Response in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<cdisclibrary>
    <products>
        <_links>
            <self>
                <href>/mdr/products</href>
                <title>CDISC Library Product List</title>
                <type>CDISC Library Product List</type>
            </self>
            <data-collection>
                <_links>
                    <self>
                        <href>/mdr/products/DataTabulation</href>
                        <title>Product Group Data Tabulation</title>
                        <type>CDISC Library Product Group</type>
                    </self>
                    <cdash>
                        <href>/mdr/cdash/1-1</href>
                        <title>Clinical Data Acquisition Standards Harmonization Model Version 1.1</title>
                    </cdash>
                </_links>
            </data-collection>
        </_links>
    </products>
</cdisclibrary>
```

The CDISC Library

REST API – Media Types

API category (highlighted rows) or specifics	JSON application/json	XML application/xml	CSV text/csv	Excel application/vnd.ms-excel
Statuses	✓			
Searches	✓			
default	✓	✓		
<p> ⓘ Referring to the category in the API Documentation</p>				
Controlled Terminology (CT)				
• /mdr/ct/packages/{product}	✓	✓	✓	✓
• All other CT endpoints	✓	✓		
Clinical Data Acquisition Standards Harmonization (CDASH)				
• /mdr/cdash/{version}	✓	✓	✓	✓
• All other CDASH endpoints	✓	✓		
CDASH Implementation Guide (CDASHIG)				
• /mdr/cdashig/{version}	✓	✓	✓	✓
• All other CDASHIG endpoints	✓	✓		
Study Data Tabulation Model (SDTM)				
• /mdr/sdtm/{version}	✓	✓	✓	✓
• All other SDTM endpoints	✓	✓		
SDTM Implementation Guide (SDTMIG)				
• /mdr/sdtmig/{version}	✓	✓	✓	✓
• All other SDTMIG endpoints	✓	✓		

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

The CDISC Library

What is a REST API?

API - Application Programming Interface

- Simplifies programming by abstracting the underlying implementation and only exposing objects or actions the developer needs.
- Defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc.
- Through information hiding, APIs enable modular programming, allowing users to use the interface independently of the implementation.

The CDISC Library

What is a REST API?



REST – REpresentational State Transfer

- An architectural style for providing data exchange standards between computer systems on the web - over HyperText Transfer Protocol (HTTP)
 - making it easier for systems to communicate with each other
- RESTful systems, are characterized by how they use a uniform and predefined set of stateless operations and separate the concerns of client and server.
- First presented by Roy Fielding in 2000 in a famous dissertation
https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- REST has become the standard for interacting with independent systems across the web

The CDISC Library

What is a REST API? - Guiding Principles

- Client-Server – separation of concerns
- Stateless Server - Each request from client to server must contain all of the information necessary to understand the request
- Cacheable - If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests. This can improve efficiency.
- Uniform Interface - REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state (HATEOAS).
- Layered System - components are grouped, i.e., layered, in a hierarchical arrangement
- Code on Demand - REST allows client functionality to be extended on the client

The CDISC Library

What is a REST API? - Anatomy

- REST uses a resource identifier (**URLs**) to identify particular resources

{scheme}:// {host}/{basePath}/{endpoint} /{Path Parameter}

`https://library.cdisc.org/api/mdr/ct/packages/{package}`

- The **GET** method is used for retrieving the information

The CDISC Library

What is a REST API? - Anatomy

- HATEOAS - Hypermedia as the Engine of Application State
- A REST client enters a REST application through a simple fixed URL. All future actions the client may take are discovered within resource representations returned from the server.
- Thus, the response message returned from the first request contains links to create further requests.

The CDISC Library

What is a REST API? - HATEOAS

GET https://library.cdisc.org/api/mdr/products

```
{  
  "_links": {  
    "data-analysis": { ... },  
    "data-collection": { ... },  
    "data-tabulation": {  
      "_links": {  
        "sdtm": [  
          {  
            "href": "/mdr/sdtm/1-2",  
            "title": "Study Data Tabulation Model Version 1.2",  
            "type": "Foundational Model"  
          }  
        ]  
      }  
    }  
  }  
}
```

GET https://library.cdisc.org/api/mdr/sdtm/1-2

```
{  
  "_links": {  
    "self": { ... }  
  },  
  "classes": [  
    {  
      "_links": {  
        "parentProduct": { ... },  
        "self": {  
          "href": "/mdr/sdtm/1-2/classes/GeneralObservations",  
          "title": "General Observation Class",  
          "type": "Class"  
        }  
      }  
    }  
  ]  
}
```

The CDISC Library

What is a REST API? - Request and Response

Request:

- URI
- Method (GET)
- Request Header
- Body



Response:

- Resource
- Response Header
- Status Code



Server

The CDISC Library

What is a REST API? - Request and Response

- Headers define the operating parameters of an HTTP transaction
- Header fields are colon-separated key-value pairs in text format

Request examples:

Accept: application/json
Accept-Charset: utf-8
Accept-Language: en-US
Host: library.cdisc.org
User-Agent: SAS/9
Connection: Keep-Alive

Response examples:

Status: 200 OK
Date: Fri, 16 Apr 2021 02:04:04 GMT
Content-Type: application/json
Content-Length: 300
Connection: Keep-Alive

The CDISC Library

What is a REST API? - Response Status Codes

- Status codes are issued by a server in response to a client's request made to the server.
 - 1xx informational response – the request was received, continuing process
 - 2xx successful – the request was successfully received, understood, and accepted (**200 OK**)
 - 3xx redirection – further action needs to be taken in order to complete the request
 - 4xx client error – the request contains bad syntax or cannot be fulfilled
(400 Bad Request, 401 Unauthorized, 404 Not Found, 408 Request Timeout)
 - 5xx server error – the server failed to fulfil an apparently valid request
(500 Internal Server Error)

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC Lua in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

The CDISC Library

REST API calls in SAS - PROC HTTP

- PROC HTTP is a powerful SAS procedure to create HTTP request

```
filename response temp;
proc http
  url="https://api.library.cdisc.org/api/mdr/products"
  out=response;
  headers
    "api-key" = "&ApiKey"
    "Accept" = "application/json";
run;

%put %sysfunc(jsonpp(response, log));

filename map temp;
libname response json map=map automap=create fileref=response;
proc copy in=response out=work;
```

The CDISC Library

REST API calls in SAS - PROC HTTP

```
NOTE: PROCEDURE HTTP used (Total process time):
      real time           0.50 seconds
      cpu time            0.04 seconds
```

```
NOTE: 200 OK
```

```
{
  "_links": {
    "data-analysis": {
      "_links": {
        "adam": [
          {
            "href": "/mdr/adam/adam-2-1",
            "title": "Analysis Data Model Version 2.1",
            "type": "Foundational Model"
          },
          {
            "href": "/mdr/adam/adam-adae-1-0",
            "title": "Analysis Data Model Data Structure for Adverse Event Analysis Version 1.0",
            "type": "Implementation Guide"
          },
          {
            "href": "/mdr/adam/adam-occds-1-0",
            "title": "ADaM Structure for Occurrence Data (OCCDS) Version 1.0",
            "type": "Implementation Guide"
          }
        ]
      }
    }
  }
}
```

```
32 proc copy in=response out=work;
33 run;
```

```
NOTE: Copying RESPONSE.ALLDATA to WORK.ALLDATA (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines. System Option for BUFSIZE was used.
INFO: Data set block I/O cannot be used because:
INFO: - The data sets use different engines, have different variables or have attributes that may differ.
NOTE: There were 611 observations read from the data set RESPONSE.ALLDATA.
NOTE: The data set WORK.ALLDATA has 611 observations and 8 variables.
NOTE: Compressing data set WORK.ALLDATA decreased size by 50.00 percent.
      Compressed is 1 pages; un-compressed would require 2 pages.
NOTE: Copying RESPONSE._LINKS_ADAM to WORK._LINKS_ADAM (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines. System Option for BUFSIZE was used.
```

16 data sets

-  Alldata
-  _links_adam
-  _links_adam2
-  _links_cdash
-  _links_cdashig
-  _links_packages
-  _links_sdtm
-  _links_sdtmig
-  _links_sdtmig2
-  _links_self
-  _links_self2
-  _links_self3
-  _links_self4
-  _links_self5
-  _links_self6
-  _links_sendig



The CDISC Library

REST API calls in SAS - PROC HTTP

- XML response

```
filename response temp;
proc http
  url="https://api.library.cdisc.org/api/mdr/products"
  out=response;
  headers
    "api-key" = "&ApiKey"
    "Accept" = "application/xml";
run;

filename map temp;
libname response xmlv2 xmlmap=map automap=replace fileref=response;
proc copy in=response out=work;
```

The CDISC Library

REST API calls in SAS - PROC HTTP

NOTE: PROCEDURE HTTP used (Total process time):

real time 0.45 seconds
cpu time 0.00 seconds

NOTE: 200 OK

```
14 filename map temp;
15 libname response xmlv2 xmlmap=map automap=replace fileref=response;
NOTE: Processing XMLMap version 2.1.
NOTE: Libref RESPONSE was successfully assigned as follows:
      Engine:      XMLV2
      Physical Name: C:\Users\LEXJAN~1\AppData\Local\Temp\SAS Temporary Files\_TD21964_WIN10PRO64_\#LN00012
16 proc copy in=response out=work;
NOTE: Writing HTML Body file: sashtml.htm
17 run;

NOTE: Copying RESPONSE._links to WORK._LINKS (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines. System Option for BUFSIZE was used.
INFO: Data set block I/O cannot be used because:
INFO:   - The data sets use different engines, have different variables or have attributes that may differ.
NOTE: There were 1 observations read from the data set RESPONSE._links.
NOTE: The data set WORK._LINKS has 1 observations and 2 variables.
NOTE: Compressing data set WORK._LINKS decreased size by 0.00 percent.
      Compressed is 1 pages; un-compressed would require 1 pages.
NOTE: Copying RESPONSE._links1 to WORK._LINKS1 (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines. System Option for BUFSIZE was used.
INFO: Data set block I/O cannot be used because:
INFO:   - The data sets use different engines, have different variables or have attributes that may differ.
NOTE: There were 1 observations read from the data set RESPONSE._links1.
NOTE: The data set WORK._LINKS1 has 1 observations and 2 variables.
NOTE: Compressing data set WORK._LINKS1 decreased size by 0.00 percent.
      Compressed is 1 pages; un-compressed would require 1 pages.
NOTE: Copying RESPONSE._links2 to WORK._LINKS2 (memtype=DATA).
```

28 data sets

 _links	 _links1
 _links2	 _links3
 _links4	 _links5
 Adam	 Adam1
 Cdash	 Cdashig
 Cdisclibrary	 Data_analysis
 Data_collection	 Data_tabulation
 Draft_content	 Packages
 Product	 Sdtm
 Sdtmig	 Sdtmig1
 Self	 Self1
 Self2	 Self3
 Self4	 Self5
 Sendig	 Terminology

The CDISC Library

REST API calls in SAS - PROC HTTP

- Excel Spreadsheet response

```
filename response "C:/_Projects/CDISC Library/response_xls/SDTMIG_3.3.xlsx";
proc http
  url="https://library.cdisc.org/api/mdr/sdtmig/3-3"
  out=response;
  headers
    "api-key"=&ApiKey
    "Accept"="application/vnd.ms-excel";
run;

proc import datafile=response
  out=work.sdtmig_3_3_tables dbms=xlsx replace;
  sheet="Datasets";
run;

proc import datafile=response
  out=work.sdtmig_3_3_columns dbms=xlsx replace;
  sheet="Variables";
run;
```

The CDISC Library

REST API calls in SAS - PROC HTTP

- Beginning with SAS 9.4M5, PROC HTTP sets up macro variables after execution:
 - `SYS_PROCHTTP_STATUS_CODE`
stores the status code of the HTTP request
 - `SYS_PROCHTTP_STATUS_PHRASE`
stores the descriptive phrase that is associated with the status code
- Use these macro variables to determine the action based on the response

The CDISC Library

REST API calls in SAS - PROC HTTP

```
%macro prochttp_check_return(code);
  %if %symexist(SYS_PROCHTTP_STATUS_CODE) ne 1 %then %do;
    %put ERROR: Expected &code., but a response was not received from the HTTP Procedure;
    %abort;
  %end;
  %else %do;
    %if &SYS_PROCHTTP_STATUS_CODE. ne &code. %then %do;
      %put ERROR: Expected &code., but received &SYS_PROCHTTP_STATUS_CODE.
&SYS_PROCHTTP_STATUS_PHRASE.;
      %abort;
    %end;
  %end;
%mend prochttp_check_return;
```

```
%prochttp_check_return(200);
```

```
NOTE: PROCEDURE HTTP used (Total process time):
      real time          0.06 seconds
      cpu time          0.01 seconds
```

NOTE: 404 Not Found

ERROR: Expected 200, but received 404 Not Found

ERROR: Execution terminated by an ZABORT statement.



The CDISC Library

REST API calls in SAS - PROC HTTP

endpoint	# SAS data sets	# SAS data sets
	JSON	XML
mdr/products	16	28
mdr/ct/packages/sdtmct-2021-03-26	8	7
mdr/sdtmig/3-3	28	32
mdr/sdtm/1-7	39	46

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

The CDISC Library

PROC LUA in SAS

- Base SAS® 9.4 introduces the LUA procedure which embeds Lua 5.2
- Lua was created in 1993 by Roberto Ierusalimschy (Pontifical Catholic University of Rio de Janeiro)
- Lua is a modern programming language with a very simple syntax
- Lua supports highly flexible data structures and modules
- Tables are the only data structure available in Lua which can be used to create different types like arrays and dictionaries
- Lua is known for excellent performance, both in speed and memory
- Lua does not replace the SAS DATA step or procedures but enhances the ability to drive SAS. Lua has direct access to the vast majority of SAS functions
- Lua makes parsing JSON very easy with one of the available modules for encoding and decoding JSON

The CDISC Library

PROC LUA in SAS

- PROC LUA runs the Lua virtual machine inside the SAS process
- Execute Lua code within a SUBMIT / ENDSUBMIT block in PROC LUA
- Lua is a dynamically typed language – variables do not have types; only values do
- Basic types in Lua:
nil, boolean, number, string, function, userdata, thread, and table

```
proc lua ;  
submit;  
  
-- Lua statements in SAS  
print('Hello world')  
  
endsubmit;  
run;
```

[Driving SAS® with Lua](#) - Paul Tomas, SGF 2015

The CDISC Library

PROC LUA in SAS

- Tables are the sole data-structuring mechanism in Lua
- They can be used to represent ordinary arrays, associative arrays, lists, symbol tables, sets, records, graphs, trees, etc.

```
-- create a table as an array
products = {'sdtm', 'sdtmig', 'sendig'}

local product = products[1]
                    |
-- loop over the array
for i, product in ipairs(products) do
| print(i, product)
end
```

1	sdtm
2	sdtmig
3	sendig

The CDISC Library

PROC LUA in SAS

```
-- create a table as a hash table of tables
local extract_attributes = {
  model =
    { template_tables = 'tmplts.tabulation_columnclassgroup',
      template_columns = 'tmplts.tabulation_columnclass'},
  ig =
    {[ template_tables = 'tmplts.tabulation_table',
      template_columns = 'tmplts.tabulation_column'}
  }
}

local model_tables = extract_attributes.model.template_tables
-- or
local ig_columns = extract_attributes['ig']['template_columns']
```

The CDISC Library

PROC LUA in SAS

```
-- loop over the tables
for i, table in pairs(extract_attributes) do
  for key, attribute in pairs(table) do
    print(i, key, attribute)
  end
end
```

model	template_tables	tmplts.tabulation_columnclassgroup
model	template_columns	tmplts.tabulation_columnclass
ig	template_tables	tmplts.tabulation_table
ig	template_columns	tmplts.tabulation_column

The CDISC Library

PROC LUA in SAS

- PROC LUA creates a special global Lua table called `sas`
- The `sas` table contains functions: `sas.scan`, `sas.symget`, `sas.symput`, ...

```
%let foo=conference;

proc lua;
  submit;
    local foo = sas.symget("foo")
    print("foo is ", foo) -- prints 'conference'
    sas.symput('foo','pharmasug')
  endsubmit;
run;

%put &foo; /* prints 'pharmasug' */
```

The CDISC Library

PROC LUA in SAS

- PROC LUA can submit SAS code
- An optional table parameter with key-value pairs can be made available for resolution in the block of SAS code

```
local products_dataset = "prod.products"

sas.submit([
    proc sort data=@dataset@;
        by @sort_key@;
        run;
    ], { dataset=products_dataset, sort_key="product_href" })
```

```
proc sort data=prod.products;
    by product_href;
run;
```

The CDISC Library

PROC LUA in SAS

- PROC LUA can read SAS datasets

productclass_title	productclass_href	product_type	product_title	product_href
Product Group Data Collection	/mdr/products/DataCollection	Foundational Model	Clinical Data Acquisition Standards Harmonization Model Version 1.1	/mdr/cdash/1-1
Product Group Data Collection	/mdr/products/DataCollection	Implementation Guide	Clinical Data Acquisition Standards Harmonization (CDASH) User Guid...	/mdr/cdashig/1-1-0
Product Group Data Collection	/mdr/products/DataCollection	Implementation Guide	Clinical Data Acquisition Standards Harmonization Implementation Guid...	/mdr/cdashig/2-0
Product Group Data Collection	/mdr/products/DataCollection	Implementation Guide	Clinical Data Acquisition Standards Harmonization Implementation Guid...	/mdr/cdashig/2-1
Product Group Terminology	/mdr/products/Terminology	Terminology	ADaM Controlled Terminology Package 19 Effective 2014-09-26	/mdr/ct/packages/adamct-2014-09-26
Product Group Terminology	/mdr/products/Terminology	Terminology	ADaM Controlled Terminology Package 24 Effective 2015-12-19	/mdr/ct/packages/adamct-2015-12-19

```
local dsid_prod = sas.open("prod.products")
sas.where(dsid_prod, "productclass_title = 'Product Group Terminology'"') -- Only Terminology

-- loop over all products
while sas.next(dsid_prod) do

    product_type = sas.get_value(dsid_prod, 'product_type')
    product_href = sas.get_value(dsid_prod, 'product_href')
    product = sas.scan(product_href, -1, '/')
    pubdate = string.gsub(string.sub(product, string.len(product)-9), '-', '')
    ctproduct = string.sub(product, 1, string.len(product)-11)

    ...

end -- end of products loop
sas.close(dsid_prod)
```

The CDISC Library

PROC LUA in SAS

- PROC LUA can write to SAS datasets
- Use the `sas.new_table` function to create a new (empty) data set template

```
function cdisclibrary.products (dataset_name)

local dsid

sas.new_table(dataset_name, {
  {name="productclass_title", type="C", length=200, label="Product Class Title"}, 
  {name="productclass_href", type="C", length=100, label="Product Class Link"}, 
  {name="product_type", type="C", length=100, label="Product Type"}, 
  {name="product_title", type="C", length=200, label="Product Title"}, 
  {name="product_href", type="C", length=64, label="Product Link"}})
dsid = sas.open(dataset_name, "u")

return dsid
end
```

The CDISC Library

PROC LUA in SAS

- PROC LUA can write to SAS datasets
- Typical scenario: open a new data set and write out observations in some sort of loop:

```
local dsid = cdisclibrary.products("prod.products")

for some-condition do

    local productclass_href = ...
    local product_type = ...
    local product_title = ...
    local product_href = ...

    sas.append(dsid)

    sas.put_value(dsid, "productclass_href", productclass_href)
    sas.put_value(dsid, "product_type", product_type)
    sas.put_value(dsid, "product_title", product_title)
    sas.put_value(dsid, "product_href", product_href)

    sas.update(dsid)
end

sas.close(dsid)
```

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

The CDISC Library

SAS with Lua

Several Lua modules to support extraction from the CDISC Library:

- rest.lua - a Lua module to easily interface with a REST API from SAS

From:

[REST Easier with SAS®: Using the LUA Procedure to Simplify REST API Interactions](#),
Steven Major, SAS Global Forum 2019

- json.lua – a Lua module for simple JSON encoding and decoding in pure Lua
<http://regex.info/blog/lua/json>, Jeffrey Friedl, 2010-2017
- cdisclibrary.lua –
a Lua module for converting CDISC Library JSON responses into SAS data sets
- fileutils.lua,
- utils.lua,
- stringutils.lua – Lua modules with support functions

The CDISC Library

SAS with Lua

rest.lua - The heart of this module is a function: rest.request

```
----- Submit a request to rest.base_url
-- ARGUMENTS:
-- action      - string - 'GET', 'POST', 'PUT', etc.
-- request     - string - the portion of the URL that follows rest.base_url
-- body_out    - string - [OPTIONAL] the fileref to write the output to. Defaults to '_bout_'
-- content_type - string - [OPTIONAL] passed to PROC HTTP's 'ct' parameter (content type)
-- header_out   - string - [OPTIONAL] the fileref to write the returned header to. Defaults to '_hout_'
-- body_in      - string - [OPTIONAL] fileref for the request body.
-- header_in    - string - [OPTIONAL] fileref for the request header.
--RETURNS:
-- pass        - boolean - whether or not the HTTP return code was in the 200's (200 OK, 201 CREATED, etc)
-- code         - number  - the actual http return code
function rest.request( action, request, body_out, content_type, header_out, body_in, header_in )
    ...
    return pass, code
end
```

The CDISC Library

SAS with Lua

- rest.lua – example use of the `rest.request` function
- Get information from the CDISC Library about the dates when products were last updated and turn that information into a Lua table
- We can use this table to determine whether we should do a new extraction

```
{  
  "_links": {  
    "self": {  
      "href": "/mdr/lastupdated",  
      "title": "Last Update Date of CDISC Library Products",  
      "type": "About CDISC Library"  
    }  
  },  
  "data-analysis": "2020-11-11",  
  "data-collection": "2020-11-11",  
  "data-tabulation": "2021-03-30",  
  "draft-content": "2021-03-30",  
  "overall": "2021-03-30",  
  "terminology": "2021-03-30"  
}
```

The CDISC Library

SAS with Lua

- rest.lua – example use of the rest.request function

```
filename luapath ("..../lua");
filename lastupd "_lastupdated_.json";
```

```
proc lua restart;
submit;
```

```
rest = require 'rest'
json = require 'json'
```

```
rest.base_url = 'https://library.cdisc.org/api'
rest.headers="Accept=""application/json"" "api-key=""xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"" "
```

```
local pass,code = rest.request('get','mdr/lastupdated', 'lastupd')
local lastupdated = json:decode(rest.utils.read('lastupd'))
```

```
local lastupdated_table={}
for key, value in pairs(lastupdated) do
  if key ~= "_links" then lastupdated_table[key] = value end
end
```

```
endsubmit;
run;
```

```
{
  ["data-collection"]="2020-11-11"
  ["overall"]="2021-03-30"
  ["data-analysis"]="2020-11-11"
  ["terminology"]="2021-03-30"
  ["draft-content"]="2021-03-30"
  ["data-tabulation"]="2021-03-30"
}
```

The CDISC Library

SAS with Lua

- Use the resulting Lua table to determine whether we should do a new extraction

```
local response_folder = sas.symget("response_folder")
local response_file = sas.io.join(response_folder, "_products.json")
sas.filename('products', response_file)

-- If the response file does not exist or is out-of-date then retrieve it
if (not sas.fileexists(response_file)) or
   (fileutils.lastmodified('products') < lastupdated['overall']) then
  local pass,code = rest.request('get','mdr/products', 'products')
end

-- convert the JSON response file into a Lua table
cdisc_products = json:decode(rest.utils.read('products'))._links

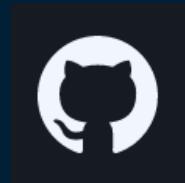
-- parse the Lua table and convert it to a SAS dataset
local dsid = cdisclibrary.products(products_dataset)
cdisclibrary.add_product_to_dataset (dsid, cdisc_products)
if dsid then sas.close(dsid) end
```

The CDISC Library

SAS with Lua

- `cdisclibrary.lua` – functions to create SAS data sets from JSON responses from CDISC Library requests for:
 - products catalog
 - Controlled Terminology packages
 - Tabulation products:
 - Foundational Models (sdtm)
 - Implementation Guides (sdtmig, sendig)

- All code can be found at GitHub:
<https://github.com/lexjansen/sas-papers/tree/master/pharmasug-2021>



The CDISC Library

SAS with Lua

Library SDTM_1_7_CLASS_TABLES SDTM_1_7_CLASS_COLUMNS

Freeze Hide Show... Format Filter... Font... Find

Table View

	name	label	description	effectiveDate	registration	version	class_name	class_label	class_description	class_structure	dataset_name
1	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	General Observations	General Observat...	The majority of observations ...		
2	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Interventions	Interventions Obs...	This SDTM class captures in...		
3	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Events	Events Observatio...	This SDTM class captures pl...		
4	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Findings	Findings Observa...	This SDTM class captures th...		
5	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Findings About	Findings About E...	This SDTM class is a speciali...		
6	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Special-Purpose	Special-Purpose ...	This SDTM class contains a ...		
7	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Trial Design	Trial Design Model	This SDTM class describes t...		
8	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Relationship	Relationship Dat...	This SDTM class provides a ...		
9	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Study Reference	Datasets for Stud...	This special purpose SDTM ...		
10	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Associated Persons	Associated Perso...	Associated Persons (AP) are ...		
11	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Special-Purpose	Demographics	A special-purpose domain th...	DM	
12	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Special-Purpose	Comments	A special-purpose domain th...	CO	
13	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Special-Purpose	Subject Element	A special-purpose domain th...	One Record per Actual ...	SE
14	SDTM v1.7	Study Data Tabul...	This document d...	2018-11-20	Final	1.7	Special-Purpose	Subject Visit	A special-purpose domain th...	One Record per Subject	SV

Library SDTM_1_7_CLASS_TABLES SDTM_1_7_CLASS_COLUMNS

Freeze Hide Show... Format Filter... Font... Find

Table View

	name	label	effectiveDate	version	class_name	dataset_name	variable_ordinal	variable_name	variable_label	variable_description	variable_simpledata	variable_role
1	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		1	STUDYID	Study Identifier	Unique identifier f...	Char	Identifier
2	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		2	DOMAIN	Domain Abbrevia...	2-character abbr...	Char	Identifier
3	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		3	USUBJID	Unique Subject I...	Identifier used to ...	Char	Identifier
4	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		4	APID	Associated Perso...	Identifier for a sin...	Char	Identifier
5	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		5	POOLID	Pool Identifier	An identifier used...	Char	Identifier
6	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		6	SPDEVID	Sponsor Device I...	Sponsor-defined i...	Char	Identifier
7	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		7	NHOVID	Non-Host Organi...	Sponsor-defined i...	Char	Identifier
8	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		8	FETUSID	Fetus Identifier	Identifier used to i...	Char	Identifier
9	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		9	FOCID	Focus of Study-S...	Identification of a...	Char	Identifier
10	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		10	-SEQ	Sequence Number	Sequence numb...	Num	Identifier
11	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		11	-GRPID	Group ID	Optional group id...	Char	Identifier
12	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		12	-REFID	Reference ID	Optional internal ...	Char	Identifier
13	SDTM v1.7	Study Data Tabulation Mo...	2018-11-20	1.7	General Observations		13	-RCFID	Invariant Record ...	Identifier for a rec...	Char	Identifier

The CDISC Library

SAS with Lua

Library SDTMIG_3_3_TABLES SDTMIG_3_3_COLUMNS

Freeze Hide Show... Format Filter... A Font... Find

Table View

	name	label	effectiveDate	version	class_name	table_ordinal	table_name	table_label	table_description	table_structure
1	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Special-Purpose	1	CO	Comments	A special-purpose domain...	One record per comment per subject
2	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Special-Purpose	2	DM	Demographics	A special-purpose domain...	One record per subject
3	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Special-Purpose	3	SE	Subject Elements	A special-purpose domain...	One record per actual Element per s...
4	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Special-Purpose	4	SM	Subject Disease ...	A special-purpose domain...	One record per Disease Milestone p...
5	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Special-Purpose	5	SV	Subject Visits	A special-purpose domain...	One record per subject per actual visit
6	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	6	AG	Procedure Agents	An interventions domain t...	One record per recorded intervention...
7	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	7	CM	Concomitant/Prio...	An interventions domain t...	One record per recorded intervention...
8	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	8	EX	Exposure	An interventions domain t...	One record per protocol-specified st...
9	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	9	EC	Exposure as Coll...	An interventions domain t...	One record per protocol-specified st...
10	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	10	ML	Meal Data	An interventions domain t...	One record per food product occur...
11	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	11	PR	Procedures	An interventions domain t...	One record per recorded procedure ...
12	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Interventions	12	SU	Substance Use	An interventions domain t...	One record per substance type per r...
13	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Events	13	AE	Adverse Events	An events domain that co...	One record per adverse event per s...
14	SDTMIG v3.3	Study Data Tabul...	2018-11-20	3.3	Events	14	CE	Clinical Events	An events domain that co...	One record per event per subject

Library SDTMIG_3_3_TABLES SDTMIG_3_3_COLUMNS

Freeze Hide Show... Format Filter... A Font... Find

Table View

	name	label	effectiveDate	table_name	column_name	column_label	column_description	column_ordinal	column_role	column_simpleData	column_core	column_codelist	column_valuelist	column_describedby
1	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	STUDYID	Study Identifier	Unique identifier for a study.	1	Identifier	Char	Req			
2	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	DOMAIN	Domain Abbreviation	Two-character abbreviation...	2	Identifier	Char	Req			CO
3	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	RDOMAIN	Related Domain Abbrevi...	Two-character abbreviation...	3	Record Qualifier	Char	Perm			
4	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	USUBJID	Unique Subject Identifier	Identifier used to uniquely i...	4	Identifier	Char	Req			
5	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	COSEQ	Sequence Number	Sequence Number given to...	5	Identifier	Num	Req			
6	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	IDVAR	Identifying Variable	Identifying variable in the p...	6	Record Qualifier	Char	Perm			
7	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	IDVARVAL	Identifying Variable Value	Value of identifying variable...	7	Record Qualifier	Char	Perm			
8	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	COREF	Comment Reference	Sponsor-defined reference ...	8	Record Qualifier	Char	Perm			
9	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	COVAL	Comment	The text of the comment. T...	9	Topic	Char	Req			
10	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	COEVAL	Evaluator	Used to describe the origin...	10	Record Qualifier	Char	Perm			
11	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	COEVALID	Evaluator Identifier	Used to distinguish multiple ...	11	Record Qualifier	Char	Perm	C96777		
12	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	COTDC	Date/Time of Comment	Date/time of comment on d...	12	Timing	Char	Perm			ISO 8601
13	SDTMIG v3.3	Study Data Tabul...	2018-11-20	CO	CODY	Study Day of Comment	Study day of the comment, ...	13	Timing	Num	Perm			
14	SDTMIG v3.3	Study Data Tabul...	2018-11-20	DM	STUDYID	Study Identifier	Unique identifier for a study.	1	Identifier	Char	Req			
15	SDTMIG v3.3	Study Data Tabul...	2018-11-20	DM	DOMAIN	Domain Abbreviation	Two-character abbreviation...	2	Identifier	Char	Req			DM
16	SDTMIG v3.3	Study Data Tabul...	2018-11-20	DM	MGNDID	Major Clinical Measure Identif...	Unique identifier for a major c...	3	Identifier	Char	Req			

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

The CDISC Library

Loading into LSAF

Standards metadata and Controlled Terminology metadata can be loaded into Life Science Analytics Framework (LSAF)

- Before this can happen some model customizations in LSAF may need to take place
 - To support newer versions
 - To support Tabulation Class Structures and Analysis Data Structures
- Some issues in the metadata may need to be resolved, for example:
 - Missing metadata
 - example: in SDTM-IG 3.2 the required LSAF submissiondatatype is missing for TR.TRMETHOD
 - Various SDTM-IG variables have multiple codelist references
 - example: in SDTM-IG 3.3 the DS.DSDECOD variable has C114118 and C66727

The CDISC Library

Loading into LSAF

Standards metadata and Controlled Terminology can be loaded into LSAF

- Variable names will need to be mapped to the names that LSAF requires
The example in GitHub has a simple Excel based mapping framework for this.

1	A	B	C	D	
	TABLE	SOURCE	TARGET	ACTION	VALUE
80	CODELIST	EFFECTIVEDATE	RELEASEDATE	RENAME	
81	CODELIST		SOURCE	ASSIGN	"NCI Thesaurus"
82	CODELIST	VERSION	SOURCEVERSION	RENAME	
83	CODELIST	REGISTRATIONSTATUS		DROP	
84	CODELIST	HREF		DROP	
85	CODELIST	CODELIST_SUBMISSIONVALUE	CODELIST_SHORTNAME	RENAME	
86	CODELIST	CODELIST_NAME	CODELIST_NAME	RENAME	
87	CODELIST	CODELIST_DEFINITION	CODELIST_DESCRIPTION	RENAME	
88	CODELIST	CODELIST_CONCEPTID	CODELIST_CODE	RENAME	
89	CODELIST		CODELIST_DATATYPE	ASSIGN	"text"
90	CODELIST		CODELIST_SASFORMATNAME	ASSIGN	
91	CODELIST	CODELIST_PREFERRED_TERM	CODELIST_PREFERRED_TERM	RENAME	
92	CODELIST	CODELIST_SYNONYMS		DROP	
93	CODELIST	CODELIST_EXTENSIBLE	CODELIST_EXTENSIBLE	RENAME	
94	CODELIST		CODELIST_EXTENSIBLE_STUDY	ASSIGN	CODELIST_EXTENSIBLE
95	CODELIST		CODELIST_SUBSETTABLE	ASSIGN	"Yes"

```
%map_extract_to_lsaf(  
  mappingds=maps.mapping,  
  tabletype=codelist,  
  template=tplts.terminology,  
  source=extract.sdtmct_20200327,  
  target=ct.sdtmct_20200327  
)
```

- The CDISC Library
- REST API
- HTTP requests from SAS
- PROC LUA in SAS
- CDISC Library example requests in SAS / Lua
- Life Science Analytics Framework (LSAF)
- Conclusion

Conclusion

- SAS fully supports extraction from the CDISC Library with PROC HTTP
- PROC LUA greatly enhances SAS/Base capabilities



References & Suggested Reading

[REST Just Got Easy with SAS® and PROC HTTP](#)

Joseph Henry, SAS Global Forum 2020

[The ABCs of the HTTP Procedure](#)

Joseph Henry, SAS Global Forum 2019

[Driving SAS® with Lua](#)

Paul Tomas, SAS Global Forum 2015

[REST Easier with SAS®: Using the LUA Procedure to Simplify REST API Interactions](#)

*Steven Major, SAS Global Forum 2019
(code on [GitHub](#))*

The Programming Language Lua (<http://www.lua.org>)
Lua.org (2013)

sas.com



Thank You ! Questions ?



sas.com

Lex Jansen

Principal Solution Consultant
Health and Life Sciences

100 SAS Campus Dr
Cary, NC 27513 USA

T +1 919 531 9860

lex.jansen@sas.com

