

Generating .xpt files with SAS, R and Python

Todd Case and YuTing Tian
Vertex Pharmaceuticals, Boston, USA

Todd Case

A Sr Director at Vertex Pharmaceuticals, where he leads the Renal Disease Area, Standards and Strategic Outsourcing teams within Biometrics. He previously led Programming teams to regulatory filings and is an invited Keynote Speaker, Requested presenter and Panelist as well as author of many papers and presentations.

Generating .xpt files with SAS, R and Python

Todd Case and YuTing Tian
Vertex Pharmaceuticals, Boston, USA

Yuting Tian

She is interested in exploring researches in statistics. She is currently working as a senior statistical programmer at Vertex Pharmaceutical Incorporated in Boston area. She presented papers with SAS and R in the SAS Global Forum 2019, JSM 2020 as well as local SAS Users Conferences.

Abstract

“CREATING SIMPLIFIED TS.XPT FILES” published by FDA, November, 2019.

The purpose for FDA guideline document:

- To explain how to create .xpt files with R and Python.

The purpose of this paper:

- To lay out a process of generating a simplified Transport (.xpt) file using SAS, R and Python
- To compare .xpt transport files created from three different languages.

Introduction

Using SAS to generate final datasets in .xpt format

- Export TS.SAS with .xpt format
- Using macro to create .xpt files with SAS

Using RStudio to generate final datasets in xpt format

- Export TS.R file with .xpt format
- A fast way to create All .xpt files with RStudio

Using Python to generate final datasets in .xpt format

- Export TS.py file with .xpt format

Review and Compare generated ts.xpt file in SAS Universal Viewer

Using SAS to generate final datasets in .xpt format

Trial summary domain:

- A trial-level dataset.
- Each record contains the value of a parameter or characteristic of the trial.
- Record basic information
- It contains both planned and actual aspects of the trial

Using SAS to generate final datasets in .xpt format

The fabricated Trial Summary(TS) raw dataset is produced shown as below:

[illegible]

Trial summary domain

STUDYID	DOMAIN	TSSEQ	TSPARMCD	TSPARM	TSVAL	TSVALNF	TSVALCD	TSVCDREF	TSVCDVER
001	TS	1	ACTSUB	Actual Number of Subjects	10				
001	TS	1	ADAPT	Adaptive Design	N		C49487	CDISC	2019-12-20
001	TS	1	AGEMAX	Maximum Age of Subjects	P65Y			ISO 8601	
001	TS	1	AGEMIN	Minimum Age of Subjects	P18Y			ISO 8601	
001	TS	1	DCUTDESC	Data Cutoff Description	11/26/2020				
001	TS	1	DOSE	Dose per Administration	400				
001	TS	1	DOSE	Dose per Administration	15				
001	TS	1	DOSFRM	Dose Form	TABLET		C42998	ISO 8601	
001	TS	1	DOSFRM	Dose Form	CAPSULES		C42998	ISO 8601	

Using SAS to generate final datasets in .xpt format

Export it with xpt format:

```
libname sasfile "E:\users\tiany";  
  
libname xptfile XPORT "E:\users\tiany\ts.xpt";  
  
data xptfile ts;  
  set sasfile.ts2;  
run;  
  
/*or another way to xport*/  
  
proc copy in=sasfile  
  out=xptfile memtype=data;  
  select ts2;  
run;
```

This PC ▸ (E:) ▸ users ▸ tiany

Name	Date modified	Type	Size
 ts	11/26/2020 3:10 PM	SAS Xport Transpo...	10 KB

2

4

1

1

1

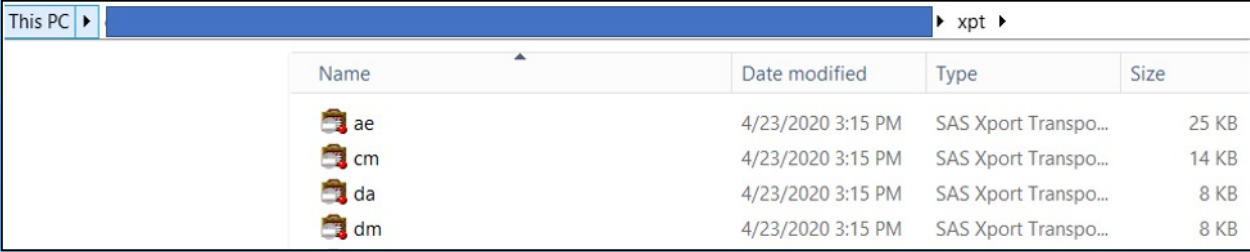
Using SAS to generate final datasets in .xpt format

A useful macro to create xpt files with SAS

```
%createxpt(inlib=sdtm, xptdir=.\xpt);
```

This macro includes two required parameters:

- inlib=XX, the name of the SAS library containing the input dataset
- xptdir is the folder where the .xpt files will be created



The screenshot shows a Windows File Explorer window with the address bar set to 'This PC' and the current directory is 'xpt'. The window displays a list of four files, each with a file icon, name, date modified, type, and size.

Name	Date modified	Type	Size
 ae	4/23/2020 3:15 PM	SAS Xport Transpo...	25 KB
 cm	4/23/2020 3:15 PM	SAS Xport Transpo...	14 KB
 da	4/23/2020 3:15 PM	SAS Xport Transpo...	8 KB
 dm	4/23/2020 3:15 PM	SAS Xport Transpo...	8 KB

Using RStudio to generate final datasets in xpt format

RStudio is a commercial product:

- An integrated development environment as a tool for statistical computing and graphics

We use RStudio as one of the programming languages:

- Create raw TS dataset
- Create SDTM.TS dataset
- Export it with xpt format.

Using RStudio to generate final datasets in xpt format

Generate raw TS domain in RStudio

##option 1 package##

Install.packages('SASxport')

1

library(SASxport)

2

library(Hmisc)

Library(sas7bdat)

3

##option 2 package##

Install.packages('haven')

4

library(Hmisc)

library(haven)

1: Install the package “SASxport”.

2: Invoke library(SASxport), library(Hmisc).

3: library(sas7bdat) to read SAS datasets into R.

4: the second way is to use “haven” package to read SAS file into R.

Using RStudio to generate final datasets in xpt format

Method 1

data.frame function

```
ts<-data.frame(STUDYID = c("001","001","001","001","001","001","001","001","001","001"),
  DOMAIN      = "TS",
  TSSEQ       = c(1,1,1,1,1,1,1,1,1,1),
  TSGRPID     = c("", "", "", "", "", "group1,drug1", "group2,drug2", "drug1", "drug2"),
  TSPARMCD    = c("ACTSUB","ADAPT","AGEMAX","AGEMIN",
    "DCUTDESC","DCUTDTC", "DOSE","DOSE","DOSFRM", "DOSFRM"),
  TSPARM      = c("Actual Number of Subjects",
    "Adaptive Design",
    "Maximum Age of Subjects",
    "Minimum Age of Subjects",
    "Data Cutoff Description",
    "Data Cutoff Date",
    "Dose per Administration",
    "Dose per Administration",
    "Dose Form",
    "Dose Form"),
  TSVAL       = c("10","N","P65Y","P18Y",
    "DATABASE LOCK",
    strftime(as.Date("2020-11-26",format="%Y-%m-%d")),
    "400", "15",
    "TABLET","CAPSULES"),
  TSVALNF     = c(""),
  TSVALCD     = c("", "C49487", "", "", "", "", "C42998", "C42998"),
  TSVCDREF    = c("", "CDISC", "ISO 8601", "ISO 8601", "", "", "", "ISO 8601", "ISO 8601"),
  TSVCDVER    = c("", strftime(as.Date("2019-12-20",format="%Y-%m-%d")), "", "", "",
    "", "", "", "", stringsAsFactors = FALSE)
```

View(ts)

Method 2

Library(haven)

```
ts<-read_sas("E:\\users\\tiany\\ts2.sas7bdat", NULL)
```

##another way##

Library(sas7bdat)

```
ts<-read.sas7bdat("E:\\users\\tiany\\ts2.sas7bdat", NULL)
```

View(ts)

1

2

1: read_sas function to read SAS datasets into R;

2: read.sas7bdat function to import SAS datasets into R.

Using RStudio to generate final datasets in xpt format

label function to assign labels for each variable

```
label(ts) <- 'Trial Summary'  
label(ts$STUDYID) <- 'Study Identifier'  
label(ts$DOMAIN) <- 'Domain Abbreviation'  
label(ts$TSSEQ) <- 'Sequence Number'  
label(ts$TSGRPID) <- 'Group ID'  
label(ts$TSPARMCD) <- 'Trial Summary Parameter Short Name'  
label(ts$TSPARM) <- 'Trial Summary Parameter'  
label(ts$TSVAL) <- 'Parameter Value'  
label(ts$TSVALNF) <- 'Parameter Null Flavor'  
label(ts$TSVALCD) <- 'Parameter Value Code'  
label(ts$TSVCDREF) <- 'Name of the Reference Terminology'  
label(ts$TSVCDVER) <- 'Version of the Reference Terminology'
```

write.xport function to export TS file with xpt format

```
write_xport(ts_file="E:/users/tiany/ts_R.xpt")
```

This PC ▸ (E:) ▸ users ▸ tiany			
Name	Date modified	Type	Size
 ts_R	11/29/2020 2:20 PM	SAS Xport Transpo...	4 KB

Using RStudio to generate final datasets in xpt format

A fast way to create ALL xpt files in a directory with RStudio

```
##import all files##  
Library(haven)  
setwd("E:\\Users\\Tiany")  
ldf<- list()  
listsas <- dir(pattern = "*.sas7bdat")  
for (k in 1:length(listsas)){  
  ldf[[k]] <- read_sas(listsas[k])  
  
  m<-substr(listsas[[k]],1,2)  
  path<-paste("E:\\Users\\Tiany\\sas\\"_m,"_new.xpt",sep="")  
  write_xpt(ldf[[k]],path)  
}
```

1: create a list

2: create the list of all the sas7bdat files in the directory

3: loop function.

4: concatenate the string and create the path

5: export all xpt files.

Using RStudio to generate final datasets in xpt format

A fast way to create ALL xpt files in a directory with RStudio

This PC ▶ (E:) ▶ users ▶ tiany ▶ sas

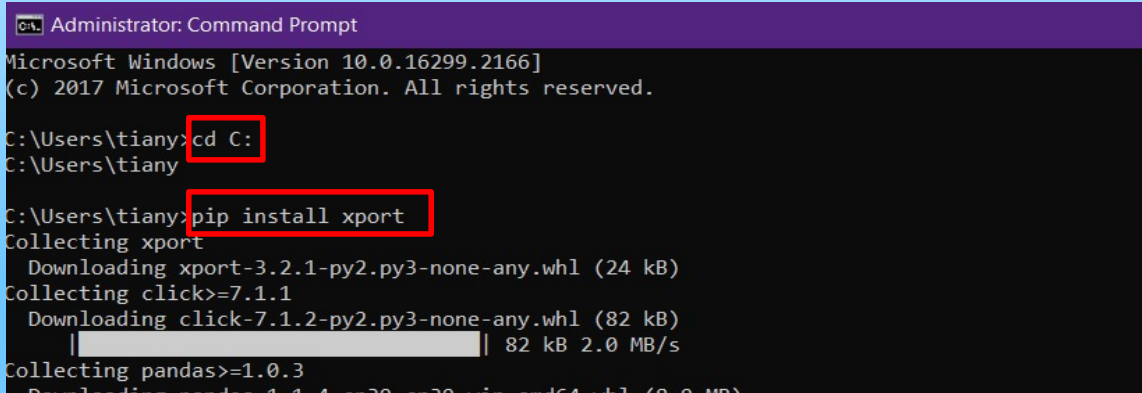
Name	Date modified	Type	Size
 ts_R	12/15/2020 5:09 PM	SAS Xport Transpo...	22 KB
 ds_R	12/15/2020 5:09 PM	SAS Xport Transpo...	32 KB
 dm_R	12/15/2020 5:09 PM	SAS Xport Transpo...	17 KB
 cm_R	12/15/2020 5:09 PM	SAS Xport Transpo...	6 KB
 ae_R	12/15/2020 5:09 PM	SAS Xport Transpo...	20 KB

Using Python to generate final datasets in xpt format

Python is an interpreted, object-oriented, high-level programming language.

Installing python, open the CMD.exe command line interface

Set the directory where python is installed and type in pip install xport



```
C:\> Administrator: Command Prompt
Microsoft Windows [Version 10.0.16299.2166]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\tiany> cd C:
C:\Users\tiany>

C:\Users\tiany> pip install xport
Collecting xport
  Downloading xport-3.2.1-py2.py3-none-any.whl (24 kB)
Collecting click>=7.1.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
    | 82 kB 2.0 MB/s
Collecting pandas>=1.0.3
  Downloading pandas-1.1.4-cp38-cp38-win-amd64.whl (6.0 MB)
```

Using Python to generate final datasets in xpt format

Generate raw TS domain in Python

```
>>> import xport
```

1

```
>>> import pandas as pd
```

2

Open the IDLE application, then start to create a new file

Import two packages, “XPORT” and “PANDAS”.

Using Python to generate final datasets in xpt format

Generate raw TS domain in Python

```
>>> ts_frame=  
pd.DataFrame(  
    {  
        "STUDYID": ["001","001","001","001","001","001","001","001","001","001"],  
        "DOMAIN": ["TS","TS","TS","TS","TS","TS","TS","TS","TS","TS"],  
        "TSSEQ": [1,1,1,1,1,1,1,1,1,1],  
        "TSGRPID": [""," "," "," "," ","group1","drug1","group2","drug2","drug2"],  
        "TSPARMCD": ["ACTSUB","ADAPT","AGEMAX","AGEMIN","DCUTDESC",  
                    "DCUTDTC","DOSE","DOSE","DOSFRM","DOSFRM"],  
        "TSPARM": ["Actual Number of Subjects",  
                  "Adaptive Design",  
                  "Maximum Age of Subjects", "Minimum Age of Subject",  
                  "Data Cutoff Description", "Data Cutoff Date",  
                  "Dose per Administration",  
                  "Dose per Administration",  
                  "Dose Form",  
                  "Dose Form"],  
        "TSVAL": ["10","N","P65Y","P18Y","DATABASE LOCK",  
                 "2020-11-26","400","15","TABLET","CAPSULES"],  
        "TSVALNF": [" "," "," "," "," "," "," "," "," "," "],  
        "TSVALCD": ["","C49487"," "," "," "," "," "," "," "],  
        "TSVCDREF": ["","CDISC","ISO 8601","ISO 8601"," "," "," ","ISO 8601","ISO 8601"],  
        "TSVCDVER": ["","2019-12-20"," "," "," "," "],  
    }  
)  
  
>>> pd.set_option("display.max_columns", None)  
>>> pd.set_option("display.max_rows", None)  
>>> ts_frame.head()
```

1

1: Date frame function to create column and row

2: set_option function to display the table

3: head function to show TS data

2

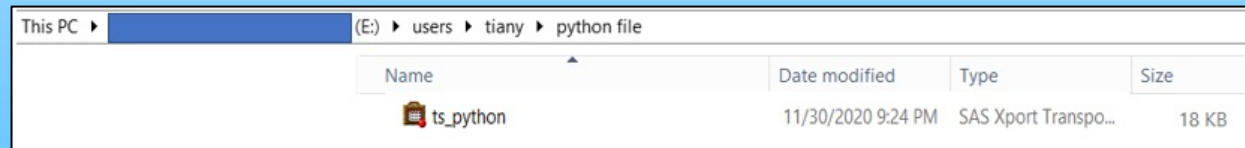
3

Using Python to generate final datasets in xpt format

Export TS.py file with .xpt format

```
>>> with open("E:/users/tiany/python file/ts_python.xpt", "wb") as f: xport.from_columns(ts_frame, f)
```

- Use (with open) function to export ts_frame.py with ts_python.xpt format
- Store it to the location "E:/users/tiany/python file"

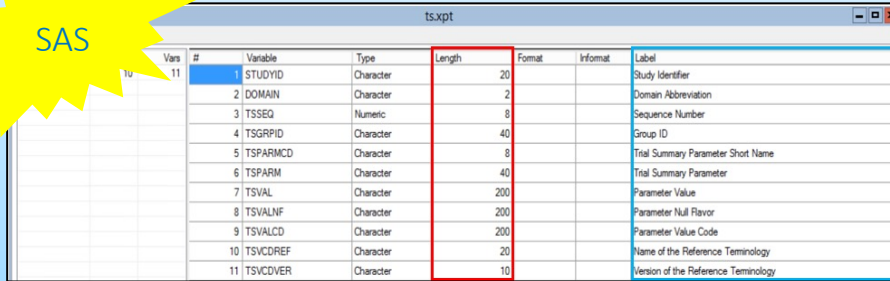


This PC ▸ (E:) ▸ users ▸ tiany ▸ python file			
Name	Date modified	Type	Size
 ts_python	11/30/2020 9:24 PM	SAS Xport Transpo...	18 KB

Review and Compare generated simplified ts.xpt file in SAS Universal Viewer

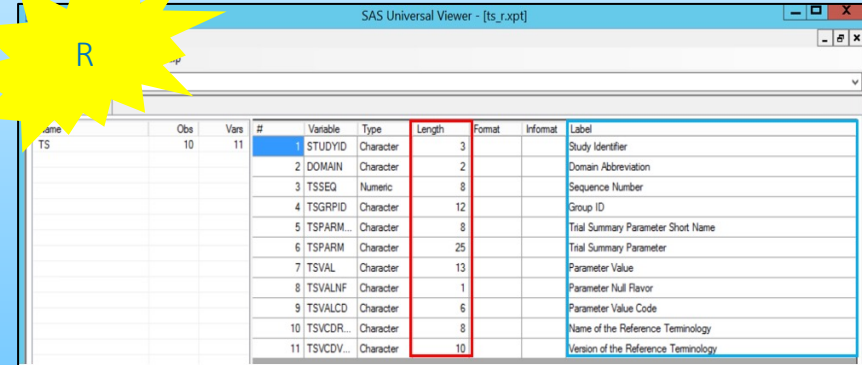
SAS Universal Viewer is to open the ts.xpt file that we created

SAS



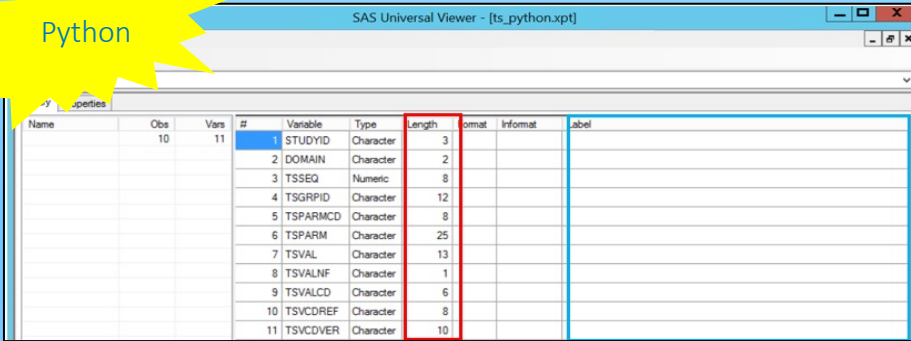
Vars	#	Variable	Type	Length	Format	Informat	Label
	1	STUDYID	Character	20			Study Identifier
	2	DOMAIN	Character	2			Domain Abbreviation
	3	TSSEQ	Numeric	8			Sequence Number
	4	TSGRPID	Character	40			Group ID
	5	TSPARMCD	Character	8			Trial Summary Parameter Short Name
	6	TSPARM	Character	40			Trial Summary Parameter
	7	TSVAL	Character	200			Parameter Value
	8	TSVALNF	Character	200			Parameter Null Flavor
	9	TSVALCD	Character	200			Parameter Value Code
	10	TSVCDREF	Character	20			Name of the Reference Terminology
	11	TSVCDVER	Character	10			Version of the Reference Terminology

R



Name	Obs	Vars	#	Variable	Type	Length	Format	Informat	Label
TS	10	11	1	STUDYID	Character	3			Study Identifier
			2	DOMAIN	Character	2			Domain Abbreviation
			3	TSSEQ	Numeric	8			Sequence Number
			4	TSGRPID	Character	12			Group ID
			5	TSPARM	Character	8			Trial Summary Parameter Short Name
			6	TSPARM	Character	25			Trial Summary Parameter
			7	TSVAL	Character	13			Parameter Value
			8	TSVALNF	Character	1			Parameter Null Flavor
			9	TSVALCD	Character	6			Parameter Value Code
			10	TSVCDR	Character	8			Name of the Reference Terminology
			11	TSVCDV	Character	10			Version of the Reference Terminology

Python



Name	Obs	Vars	#	Variable	Type	Length	Format	Informat	Label
	10	11	1	STUDYID	Character	3			
			2	DOMAIN	Character	2			
			3	TSSEQ	Numeric	8			
			4	TSGRPID	Character	12			
			5	TSPARMCD	Character	8			
			6	TSPARM	Character	25			
			7	TSVAL	Character	13			
			8	TSVALNF	Character	1			
			9	TSVALCD	Character	6			
			10	TSVCDREF	Character	8			
			11	TSVCDVER	Character	10			

Review and Compare generated simplified ts.xpt file in SAS Universal Viewer

- Firstly, when we look at ts.xpt file generated by SAS, the length of variables is corresponding with the standard variables' length in SDTM specification document. SAS allows us to change the variables' length. However, in both R and Python, the variables' length depends on the length of elements in a vector.
- Secondly, when we look at the “label” column, both SAS and R can assign variables' labels, however, it does not work in Python because the Python *Xport Module* does not have advanced properties capabilities such as variable labels cannot be assigned.
- In conclusion, SAS and R include more advanced capabilities to edit .xpt file

Thank you!

Contact Information

Todd Case: Todd_Case@vrtx.com

YuTing Tian: YuTing_Tian@vrtx.com