



FRONTAGE

YOUR DRUG DEVELOPMENT PARTNER



DMPK



Bioanalytical



CMC



Clinical



Biometrics



Creating Simplified TS domain Using R Shiny App

Hanming Tu
VP of Clinical IT and Database Administration
Frontage Laboratories, Inc



Agenda

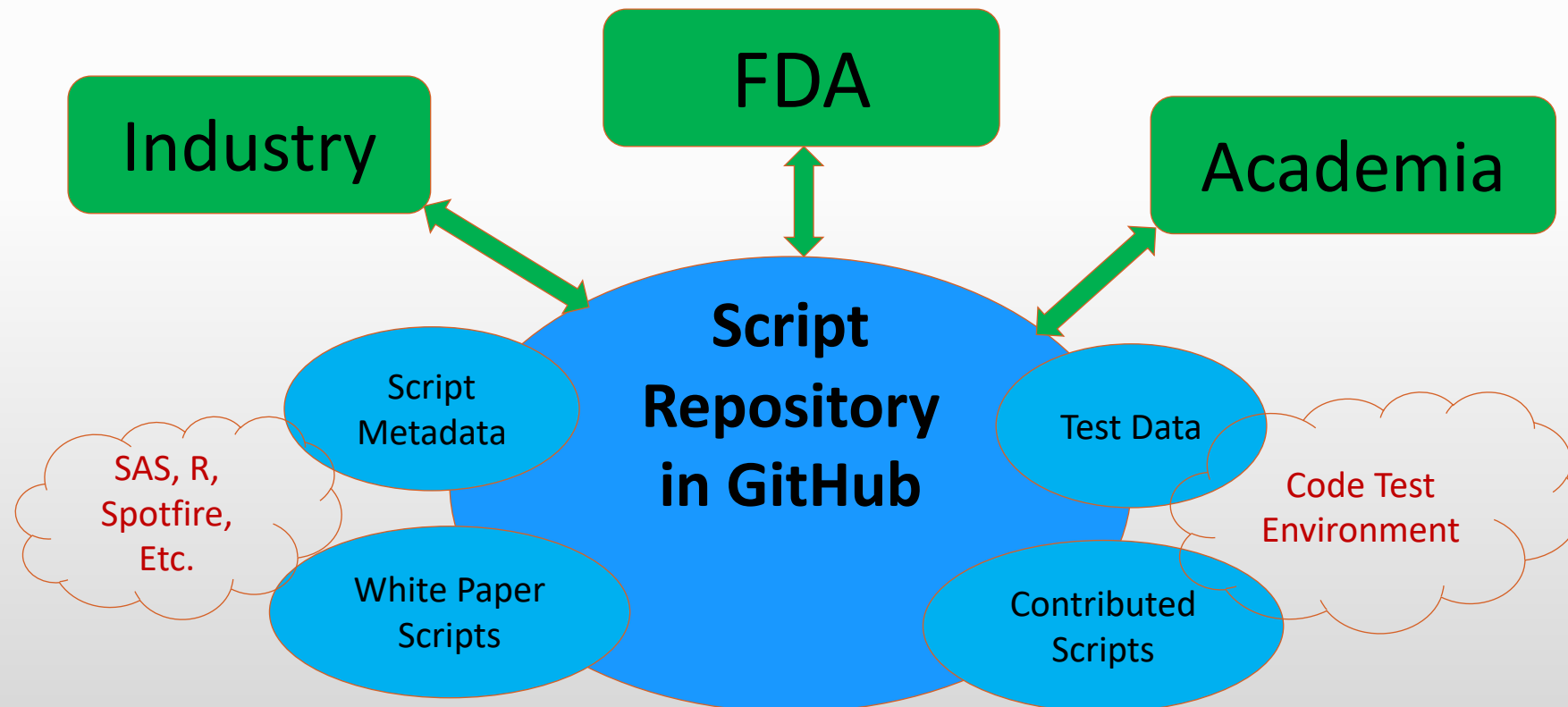
- Project Background
- Why We Need Simplified TS Domain?
- What Tool We Need?
- How We Develop a Tool for TS Domain?
- Demo



Background: Code Sharing Repository

PHUSE: Standard Analyses and Code Sharing Working Group

Co-Leads: Hanming Tu, Frontage Laboratories, Inc; Bob Friedman, Xybion; Wendy Dobson, PHUSE Project Manager



Vision: Collaboration Platform and Shared Reusable Code Library





Why Simplified TS Domain?

- Study Data Technical Conformance Guide
 - ✓ Provide guidance on simplified and full ts.xpt in Trial Design Model (TDM)
 - ✓ Define a list of parameters that should be submitted where relevant for clinical studies for a full ts.xpt
- Technical Rejection Criteria for Study Data
 - ✓ If a clinical study, submitted to CDER or CBER, started on or prior to December 17, 2016, for NDAs, BLAs, and ANDAs, and the study contains an xpt dataset (other than the ts.xpt), a simplified ts.xpt file should be submitted
 - ✓ If a nonclinical study, submitted to CDER, started on or prior to December 17, 2016, for NDAs, BLAs, and ANDAs (or December 17, 2017, for Commercial INDs), whether or not the study contains an xpt dataset (other than the ts.xpt), a simplified ts.xpt file should be submitted



Why Simplified TS Domain Important?

- Simplified TS Files provide a Study Start Date for NDA/BLA/ANDA studies that began on or before December 17, 2016 and Commercial IND studies that began on or before December 17, 2017
- It allows FDA to determine that study data is not required to be in a CDISC standardized format.
- Sponsors should submit a dataset named 'ts.xpt' with four variables (STUDYID, TSPARMCD, TSVAL, and TSVALNF) and one row of information.
- Additional details can be found in the Study Data Technical Conformance Guide.



Why Your Submission Is Rejected?

Error	Description
1734	Trial Summary (TS) dataset (ts.xpt) with information on study start date must be present for required sections*
1735	Correct STF file-tags must be used for all standardized datasets and corresponding define.xml files in required sections*
1736	For SEND data, a DM dataset and define xml must be submitted in required sections* For SDTM data, a DM dataset and define.xml must be submitted in required sections* For ADaM data, an ADSL dataset and define.xml must be submitted in required sections*

* Refer to the latest Technical Rejection Criteria for Study Data

➤ FDA Technical Reject Criteria

	Submission Type		Study Type			Total
	Original	Other	Nonclinical (m4)	Clinical (m5)	Other	
Total Number of Studies	591	497	N/A	1004	74	1078
Total Number Studies with Critical Errors	392	281	N/A	673	0	673
Error 1734	77	109	N/A	186	0	186
Error 1735	327	170	N/A	497	0	497
Error 1736	55	33	N/A	88	0	88
Error Rate (% among Total Number of Studies)	67.5%	56.5%	N/A	67.0%	0	62.4%



What Is TS.XPT Datasets Look Like?

- STUDYID is a unique identifier for a study within the submission in the Study Tagging File (STF).
- TSPARMCD is trial summary parameter short name.
 - ✓ SSTDTC is study start date for clinical studies; it is in ISO 8601 format
 - ✓ STSTDTC is study start date for nonclinical studies (SEND)
- TSVALNF is parameter null flavor

For a study with a valid SSD:

STUDYID	TSPARMCD	TSVAL	TSVALNF
study ID in STF	SSTDTC	yyyy-mm-dd	

For a study without a valid SSD:

STUDYID	TSPARMCD	TSVAL	TSVALNF
study ID in STF	SSTDTC		As specified in the ISO 21090 Standard

Clinical Studies

Nonclinical Studies

For a study with a valid SSD:

STUDYID	TSPARMCD	TSVAL	TSVALNF
study ID in STF	STSTDTC	yyyy-mm-dd	

For a study without a valid SSD:

STUDYID	TSPARMCD	TSVAL	TSVALNF
study ID in STF	STSTDTC		As specified in the ISO 21090 Standard



What Technical Requirements Are For a Tool?

Req No.	Requirements	Acceptance Criteria
R-1	Allow users to enter a Study ID	- Field label: "Study ID (STUDYID)" - Free text field, • Maximum 200 characters, • Format variable value as character - Field should be independent from R-2 field
R-2	Allow users to select a Study Type	- Field label: "Study Type Parameter (TSPARMCD)" - Dropdown Selections: o "Clinical (SSTDTC)"; o "Nonclinical (STSTDTC)" - Format variable value as character
R-3	Allow users to select a Study Start Date	- Field label: "Study Start Date (TSVAL)" - Include validation to format date and variable value as YYYY-MM-DD - Format variable value as character - R-3 and R-4 are mutually exclusive, i.e. If R-3 is filled, then R-4 should be blank/disabled
R-4	Allow users to select an Exception Code	- Field label: "Exception Code (TSVALNF)" - Free text field - Format variable value as character - R-4 and R-3 are mutually exclusive i.e. If R-4 is filled, then R-3 should be blank/disabled
R-5	Allow users to view variable values	- StudyID Variable Value is what the user entered in the Study ID Field (R-1) - TSPARMCD Variable Value is what the user selected in the Study Type Field, SSTDTC or STSTDTC (R-2) - TSVAL Variable Value is what the user selected in the Study Start Date Field (R-3) - TSVALNF Variable Value is what the user selected in the Exception Code Field (R-4) - Variable values are visually associated with their variable names and field labels - Variable values are formatted as they will appear in the XPT file



What Technical Requirements Are For a Tool?

Req No.	Requirements	Acceptance Criteria
R-6	Allow users to download and generate an XPT file containing the four variables entered the fields	• Button labeled: "Download" • Nonfunctional requirements for generated XPT files included in worksheet 2
R-7	Generated XPT file is titled, "ts.xpt"	N/A
R-8	Generated XPT file is in SAS Transport Format Version 5	N/A
R-9	Generated XPT file contains one dataset with the SAS name, "TS"	N/A
R-10	Generated XPT file contains four variables in one row of information: STUDYID, TSPARMCD, TSVAL, TSVALNF	• StudyID Variable Value is what the user entered in the Study ID Field (R-1) • TSPARMCD Variable Value is what the user selected in the Study Type Field, SSTDTC or STSTDTC (R-2) • TSVAL Variable Value is what the user selected in the Study Start Date Field in the format of YYYY-MM-YY (R-3) • TSVALNF Variable Value is what the user entered in the Exception Code Field (R-4)
R-11	Generated XPT file contains all variable values formatted as characters	N/A



What Enhancement Requirements Are?

Defects #/ Enhancement #	Defects / Enhancements
E-1	Study Start Date (TSVAL) and Exception Code (TSVALNF) should not be mutually exclusive. -User can add values to either fields without any inter- dependency
E-2	Default value in Study Start Date (TSVAL) field should be blank, not today's date.
E-3	"Exception Code (TSVALNF)" should be a Dropdown with only one value "NA" Note - Currently it is a free text field
E-4	In View tab, only shows one entry at a time. Remove show entry functionality.
E-5	In View tab, remove search function
E-6	In View tab, remove use of paging functionality
E-7	Edit the link text from "Creation Guide" to "Simplified ts.xpt Creation Guide" and point to the Study Data for Submission to CDER and CBER Page URL - " https://www.fda.gov/industry/study-data-standards-resources/study-data-submission-cder-and-cber "
E-8	Remove the Link "Utility Requirements"
D-1	"Study ID (STUDYID)" field should have a character limit of 200



What FDA Technical Recommendation Is?

➤ <https://www.fda.gov/media/132457/download>



**U.S. FOOD & DRUG
ADMINISTRATION**



OR



CREATING SIMPLIFIED TS.XPT FILES



How To Use R, R Package, RStudio Project?

- **R** is an open source programming language and software environment for statistical computing and graphics
- R **package** is the fundamental unit of shareable code bundled with data, tests, examples, and documentation.
- **RStudio** is a free and open-source integrated development environment (IDE) for R.
- Rstudio **project** helps you organizing your development and build of a R code or a Package.



How: R Packages Used in Development

- Package **shiny**: is an R package that makes it easy to build interactive web apps straight from R. You can also use Shiny server to set up a web server.
- Package **shinydashboard**: provides an easy way to create dashboard.
- Package **shinyjs**: lets you perform common useful JavaScript operations in Shiny apps that will greatly improve your apps without having to know any JavaScript
- Package **shinyBS**: Twitter Bootstrap Components for Shiny to add additional functionality and interactivity to your Shiny applications
- Package **SASxport**: to read and write 'SAS' 'XPORT' Files It includes functions for reading, listing the contents of, and writing 'SAS' 'xport' format files.



How: R Packages Used in Development

- Package **Hmisc**: contains many functions useful for data analysis, high-level graphics, utility operations, functions for computing sample size and power, importing and annotating datasets, variable clustering, etc.
- Package **rhandsonable**: is a htmlwidget based on the handsontable.js library. Handsontable is a data grid component with an Excel-like appearance and with powerful features like data validation, sorting, grouping, etc.
- Package **phuse**: is a web application framework for phuse-scripts repository. It provides some useful functions that allow you to search and clone a repository.
- Package **DT**: provides an R interface to the JavaScript library DataTables, providing filtering, pagination, sorting, and many other features in the tables.
- Package **V8**: is an R interface to V8: Google's open source JavaScript and WebAssembly engine.
- Package **stringr**: is simple, consistent wrappers for common string operations.



How: To Design Web Interface

- A web-based R application usually has three components: user interface, server and run-time.

```
ui <- dashboardPage( ... )  
server <- function(input, output, session) { .. }  
shinyApp(ui, server)
```

- The shinydashboard package provides an easy and simple way of laying out the header, sidebar and page of a dashboard

```
header <- dashboardHeader(title = "Creating TS Domain" )  
sidebar <- dashboardSidebar(  
  sidebarMenu(id = "tab1", menuItem("Simplified TS", icon = icon("cog")  
    , menuSubItem(), menuSubItem() ) ) )  
ui <- dashboardPage( header, sidebar, dashboardBody(useShinyjs(), ...  
  , fluidRow(tabsetPanel(id='tabs'  
    , tabPanel("Create", uiOutput("tabP1")), tabPanel("View", uiOutput("tabP2"))))  
  , tags$footer("PhUSE Code Sharing (Repository) Project"  
  )
```



How: To Control Inputs

- R provides three functions to interact with user's inputs
 - ✓ Function **reactive**: allows a user to monitor the status of an input or other changing variable and return the value to be used elsewhere in the code
 - ✓ Function **observe**: is similar reactive, the main difference is that it does not return any values to any other environment besides its own, and it is not lazy.
 - ✓ Function **observeEvent**: is an event trigger. It watches one event or change in a variable, and then fires when the event happens.

```
# toggle the sidebar
observe({ if(input$showpanel == TRUE) { js$showSidebar() } else { js$hideSidebar() } })
# whether required fields are filled for downloading
observeEvent(input$studyid, { if (input$studyid == "") ; hide("downloadData"); else show("downloadData") })
# limit study id to 200 characters
observeEvent(input$studyid, {
  if(input$studyid != "") {
    if (str_length(input$studyid)>200) { updateTextInput(session, "studyid", value = str_sub(input$studyid, end=200) )
      createAlert(session, "alert", "StudyIDAlert", title = "StudyID Alert", content = "Study ID must not exceed 200 characters and will be truncated."
        , dismiss = TRUE, style = "warning", append = FALSE); Sys.sleep(3)
    } else { closeAlert(session, "StudyIDAlert") }
  }
})
```



How: To Display Data

- Two functions are developed to store and process the variables for the TS domain table:

```
get_tsval <- reactive({ if (is_empty(input$tsval)) { tsval <- as.character("") } else { tsval <-  
as.character(input$tsval) } tsval })  
  
ts_content <- reactive({  
  validate( need(input$studyid != "", "Please provide Study ID.") )  
  validate( need(input$tsparmcd != "", "Please provide Study Type Parameter.") )  
  req(input$studyid); req(input$tsparmcd)  
  ts <-data.frame(STUDYID=input$studyid,TSPARMCD=input$tsparmcd,TSVAL=get_tsval(),TSVALNF=input$tsvalnf );  
  ts})
```

- Function datatable from DT package is used to display the context

```
output$DT2 <- renderDataTable({ df <- ts_content(); str(df); datatable(df, options = list(dom = 't')); })  
output$tabP2 <- renderUI({ tabPanel("View", DT::dataTableOutput("DT2"), hr()  
  , hidden(downloadButton('downloadData', 'Download')) ) })
```



How: To Create XPT file

- Use write.xport function from the SASxport package to convert data frame into xpt format

```
# write out dataframe
write.xport(list=aList, file = file, verbose=FALSE, sasVer="7.00", osType="R 3.5.2", cDate=Sys.time()
, formats=NULL, autogen.formats=TRUE )
```

- Use R downloadHandler for downloading the data to user's local computer

```
output$downloadData <- downloadHandler(
  filename = function() { paste("ts", ".xpt", sep="") },
  content  = function(file = filename) {studyData <- ts_content()
# 1. set to characters not factors;
# 2. Set length for character fields;
# 3. Label list ..
# 4. place this dataset into a list with a name
aList = list(studyData); names(aList)[1]<-"TS"
attr(aList,"label") <- "TRIAL SUMMARY"
# write out dataframe [see above ]
} )
```



How: To Download and/or Access the tool

- The genTS tool is embedded in phuse package as the 7th example.
- The phuse package is hosted in GitHub and accepted by Comprehensive R Archive Network (CRAN).
- The genTS application is published to shinyapps.io website.

How to get R <i>phuse</i> Package	How to run R <i>phuse</i> Package
In R or RStudio <pre>install.packages("devtools") library(devtools) install_github("TuCai/phuse")</pre> Or directly install from CRAN <pre>install.packages("phuse")</pre>	Run the application in RStudio <pre>library(phuse) start_app(n=7)</pre> Run the application from hosted website • https://geotiger.shinyapps.io/07_genTS/



Package Maintainability and License

- The application in the phuse package is developed under the Code Sharing (Repository) project in Standard Analysis and Code Sharing working group in **PHUSE** and hosted in GitHub repository and published in CRAN.
- **GitHub** is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.
- The “Comprehensive R Archive Network” (**CRAN**) is a collection of sites which carry identical material, consisting of the R distribution(s), the contributed extensions, documentation for R, and binaries.
- R is released under the GNU General Public License (**GPL**), version 2. The license must not restrict anyone from making use of the program in a specific field of endeavor.



How: To Create a TS domain file



➤ https://geotiger.shinyapps.io/07_genTS/

Creating TS Domain

Simplified TS

Show/Hide sidebar

Create View

Study ID (STUDYID) *

Study Type Parameter (TSPARMCD) *

Nonclinical (STSTDTC)

Study Start Date (TSVAL)

Exception Code (TSVALNF)

__Blank__

About phuse Pkg

Simplified ts.xpt Creation Guide

Source Code

PhUSE Code Sharing (Repository) Project



Question and Answer

