

Automating SAS Program Table of Contents for Your FDA Submission Package

Lingjiao Qi, Bharath Donthi, Statistics & Data Corporation

ABSTRACT

To submit a complete and compliant data package to the FDA for product approval, the submission must include the SAS programs that generated the analysis datasets, tables, and figures. Including these programs in the submission package helps the FDA data reviewers to understand the process by which the variables for the respective analyses were created, and to confirm the analysis algorithms. Organizing these SAS programs into a Table of Contents is highly recommended as it serves as an easy reference for the FDA reviewers, verifies that each expected file is included in the submission, and provides the FDA with easily-accessible details for each program.

The Table of Contents is usually compiled manually by reviewing each individual program and typing the required information into a word processor. This time-consuming process is inefficient and error-prone. To increase accuracy and efficiency, we have developed an in-house macro tool to automatically generate a Table of Contents by reading each submitted SAS program and its associated files. This easy-to-use macro tool can be fully executed within SAS, and it dramatically reduces documentation preparation time.

To produce a complete and detailed Table of Contents, the macro tool extracts from the submitted programs: metadata (program name, size, descriptions, etc.), input datasets, output file names, and macros. This paper will provide a detailed description of our time-saving macro tool to assist SAS users in automatically generating the Table of Contents for their FDA data submission packages.

INTRODUCTION AND BACKGROUND

Over the last decades, the US Food and Drug Administration (FDA) has released a series of guidance documents for statistical software packages. Specifically, the FDA Study Data Technical Conformance Guide (Figure 1) requests that sponsors provide software programs as part of regulatory submissions. Inclusion of these programs in the submission package facilitates the FDA data reviewers' understanding of the statistical analyses and per the guidance below should be included.

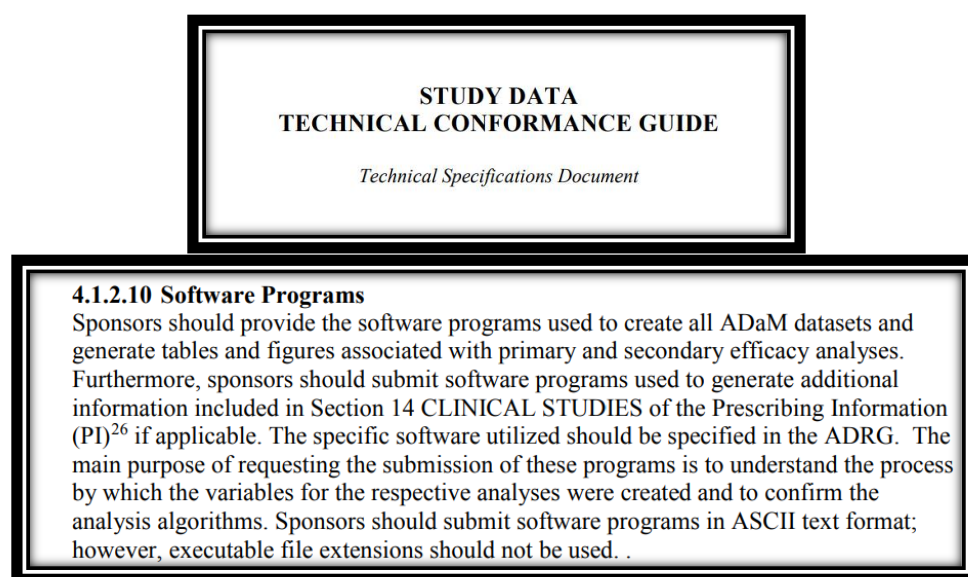


Figure 1. FDA Study Data Technical Conformance Guide Requires Software Submission

Because electronic submissions can be complex and large, providing a comprehensive list of submitted software programs helps ensure that all files that were intended to be sent were actually included. In addition, by providing clear documentation of all software programs provided to the FDA, reviewers gain road maps to the details of each program, allowing for a better understanding of the data flow. These details might include software versions, output produced, global macros called, and source SAS datasets utilized by each program. Thus, a detailed table of contents (TOC) providing descriptions of each program to be submitted and the usage of all submitted data is strongly recommended for any electronic submission.

Generating a detailed TOC is traditionally created through a labor-intensive manual process in which a programmer opens and checks each line of SAS code in every submission-ready program, transcribes the required information into a word processor, and finally composes a TOC. Examples of common mistakes arising from manual compilation range from simple spelling errors to incorrect cross-referencing of metadata, input datasets, global macro usage, or program outputs (tables, figures, etc.). Additionally, reopening and manually scanning through each submission-ready program carries the risk of accidental modification to the code requiring an extraneous round of validation. Revisiting each submission-ready program to collate and present needed information can easily amount to a full week's worth of work or longer depending on the complexity of the project. Additional quality control processes may create several cycles of revisions thus increasing time needed to create submission-ready documents.

Thus, it is highly desirable to automate the process of creating the TOC by non-invasively extracting the required information within existing submission-ready SAS programs and their associated materials. This paper introduces a macro utility to automate the construction of a detailed TOC which is executed completely within SAS, without reopening each SAS program. The logic behind the macro development is illustrated step by step with corresponding outputs provided to the reader in the subsequent sections. The macro is designed to track and extract metadata on SAS programs and their logs, determine the input source SAS datasets, document any utilized global macros, identify corresponding outputs, and finally create a completed TOC ready for FDA submission. The macro generated TOC identifies each program and its location within the submission. It also serves as cross-referenced documentation associated with all datasets, global macros, and outputs contained in an electronic submission. The macro generated TOC is also imbedded with hyperlinks routing back to the original location of SAS programs, permanent SAS datasets, and any utilized global macros to facilitate navigation to and review of outputs.

Beyond serving as an essential aid for FDA reviewers, programmers themselves may benefit from using the TOC as an extra validation tool to verify the metadata of individual programs and to track source dataset usage and corresponding outputs.

AN ACCURATE, NON-INVASIVE, AND AUTOMATIC METHOD TO GENERATE TABLE OF CONTENTS FOR SAS PROGRAMS: %PROGTOC

To eliminate the time-consuming and error-prone process mentioned above, we developed an easy-to-use macro to automatically produce a detailed TOC for all submitted SAS programs. The macro we propose requires minimal updates from users and only utilizes common base SAS procedures and base macro processing. Though study datasets and programs should be organized into a specific file directory structure when submitted in the eCTD (electronic common technical document) format to FDA (Figure 2 and Figure 3), we developed the macro to offer the flexibility to specify file pathways of physical SAS programs, outputs, etc. so that the macro can be used at any stage of programming activities where clinical datasets and programs might reside in a different location than the datasets and programs within the submission. Thus, the location of the macro relative to the location of the generated TOC is irrelevant. Besides helping the e-submission process, the generated TOC can also be used for tabulation purposes to link programs to the corresponding source datasets and associated outputs, as well as to detail ongoing metadata at any programming stage.

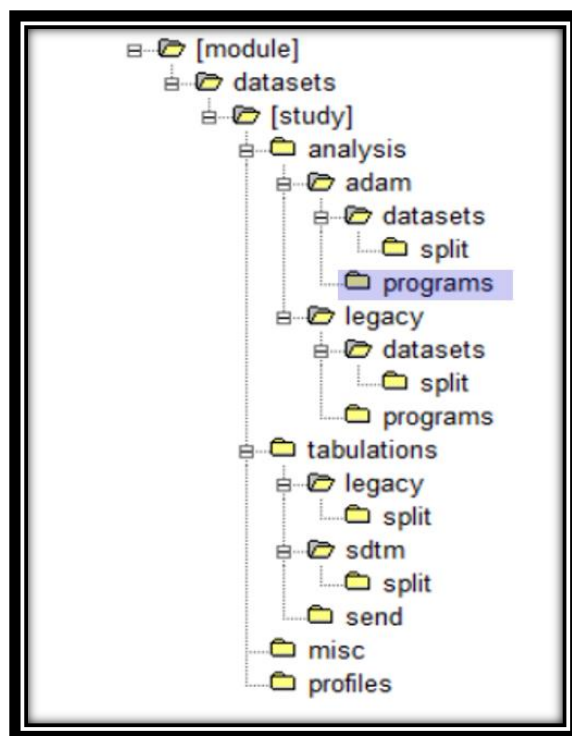


Figure 2. Folder Structure for Study Datasets and Programs Required by FDA Study Data Technical Conformance Guide

Study101 > Analysis > ADaM > Programs			
Name	Date modified	Type	Size
a_adae_sas	8/15/2018 8:21 AM	Text Document	4 KB
a_adcm_sas	8/15/2018 8:21 AM	Text Document	4 KB
a_adeq_sas	8/15/2018 8:30 AM	Text Document	5 KB
a_adlb_sas	8/15/2018 9:29 AM	Text Document	6 KB
a_admh_sas	8/15/2018 9:30 AM	Text Document	4 KB
a_adsl_sas	8/15/2018 7:59 PM	Text Document	6 KB
align_sas	5/1/2013 1:00 PM	Text Document	3 KB
filenum_sas	1/11/2018 5:33 PM	Text Document	3 KB
initialize2_sas	1/4/2018 9:01 AM	Text Document	12 KB
labllngth_sas	1/16/2018 9:45 AM	Text Document	5 KB
stat_freq_sas	1/10/2018 12:57 PM	Text Document	4 KB
stat_means_sas	4/22/2013 7:18 AM	Text Document	7 KB
t_ae1_sas	8/16/2018 3:37 PM	Text Document	4 KB
t_ae2_sas	8/15/2018 12:47 PM	Text Document	11 KB
t_ae3_sas	8/15/2018 1:05 PM	Text Document	11 KB
t_aesev_sas	8/15/2018 1:06 PM	Text Document	10 KB
t_cm_sas	8/15/2018 4:19 PM	Text Document	7 KB
t_dm_sas	8/16/2018 3:38 PM	Text Document	8 KB
t_ds_sas	8/16/2018 3:40 PM	Text Document	10 KB

Figure 3. Example of a List of SAS Programs Submitted to FDA (including ADAM, Global Macro, Table, and Figure Programs)

The macro call is shown as follows:

```
%macro PROGTOC (TXTPATH=,
                TLFPATH=,
                ADPATH=,
                MACPATH=,
                ADOUT=)
```

Parameters:

TXTPATH= pathway of submitted SAS programs in TXT format

TLFPATH= pathway of TLF program log

ADPATH = pathway of ADAM program log

MACPATH= pathway of global macro library

ADOUT = pathway of submitted ADAM datasets in XPT format

The macro generates a detailed TOC that lists all SAS program names, file size, physical location, output name, source datasets used, and global macros called within each program (Figure 4). The generated TOC displays multiple rows where each row represents one unique program for one unique output: Analysis Data Model (ADaM) dataset or Table/Figure output used in a clinical study report. If needed, this macro may be updated to include more metadata (e.g., Creation Date/Time and Last Modified Date/Time) and input/output information.

Statistics & Data Corporation
Version: FINAL

Sponsor Name.
Study: Protocol Number

Programs Submitted
Table of Contents

Program Name	Type	Class	Output Name	Input Dataset	Global Macro Used	Size (KB)	Location
a_adae_sas.txt	SAS	ADaM	Create Dataset: ADAE	ADSL, AE	INITIALIZE2, LABELLENGTH	4	C:\Study101\Analysis\ADaM \Programs\adae_sas.txt
a_adbio_sas.txt	SAS	ADaM	Create Dataset: ADBIO	ADSL, XO	INITIALIZE2, LABELLENGTH	6	C:\Study101\Analysis\ADaM \Programs\adbio_sas.txt
a_adcm_sas.txt	SAS	ADaM	Create Dataset: ADCM	ADSL, CM	INITIALIZE2, LABELLENGTH	4	C:\Study101\Analysis\ADaM \Programs\adcm_sas.txt
t_mh_sas.txt	SAS	Table	Create Table: 14.1.3.1	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM \Programs\t_mh_sas.txt
t_mh_sas.txt	SAS	Table	Create Table: 14.1.3.2	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM \Programs\t_mh_sas.txt
t_urine_sas.txt	SAS	Table	Create Table: 14.3.2.7	ADLB, ADSL	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM \Programs\t_urine_sas.txt
t_val_sas.txt	SAS	Table	Create Table: 14.3.3.1	ADSL, ADVA	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM \Programs\t_val_sas.txt
t_va2_sas.txt	SAS	Table	Create Table: 14.3.3.2	ADSL, ADVA	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM \Programs\t_va2_sas.txt
t_va2_sas.txt	SAS	Table	Create Table: 14.3.3.3	ADSL, ADVA	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM \Programs\t_va2_sas.txt
t_vs_sas.txt	SAS	Table	Create Table: 14.3.8.1	ADSL, ADVS	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM \Programs\t_vs_sas.txt

Figure 4. Example of TOC Output Generated by %PROGTOC

The cells in columns (1st, 4th, 5th, 6th) of the TOC output are imbedded with hyperlinks which route to the original file destination (Figure 5). This automated and detailed TOC ensures faster program documentation while maintaining a high level of accuracy with regard to traceability, reliability, and searchability. Thus, this macro utility helps achieve high-quality submissions in a cost-effective and efficient manner.

Programs Submitted
Table of Contents

Program Name	Type	Class	Output Name	Input Dataset	Global Macro Used	Size (KB)	Location
a_adae_sas.txt	SAS	ADaM	Create Dataset: ADAE	ADSL, AE	INITIALIZE2, LABELLENGTH	4	C:\Study101\Analysis\ADaM\Programs\a_adae_sas.txt
a_adbio_sas.txt	SAS	ADaM	Create Dataset: ADBIO	ADSL, XO	INITIALIZE2, LABELLENGTH	6	C:\Study101\Analysis\ADaM\Programs\a_adbio_sas.txt
a_adcm_sas.txt	SAS	ADaM	Create Dataset: ADCM	ADSL, CM	INITIALIZE2, LABELLENGTH	4	C:\Study101\Analysis\ADaM\Programs\a_adcm_sas.txt
t_mh_sas.txt	SAS	Table	Create Table: 14.1.3.1	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM\Programs\t_mh_sas.txt
t_mh_sas.txt	SAS	Table	Create Table: 14.1.3.2	ADMH, ADSL	INITIALIZE2, FILENUM	6	C:\Study101\Analysis\ADaM\Programs\t_mh_sas.txt
t_urine_sas.txt							C:\Study101\Analysis\ADaM\Programs\t_urine_sas.txt
t_val_sas.txt							C:\Study101\Analysis\ADaM\Programs\t_val_sas.txt
t_va2_sas.txt							C:\Study101\Analysis\ADaM\Programs\t_va2_sas.txt
t_va2_sas.txt							C:\Study101\Analysis\ADaM\Programs\t_va2_sas.txt
t_vs_sas.txt							C:\Study101\Analysis\ADaM\Programs\t_vs_sas.txt

Figure 5: Hyperlinks Imbedded in TOC Generated by %PROGTOC

LOGIC BEHIND THE SCENES

STEP 1: RETRIEVE METADATA OF SUBMITTED PROGRAMS

Per the FDA Study Data Technical Conformance Guide, submitted software programs should be in ASCII text formats and all submitted programs (ADaM programs, table and figure programs, and submitted global macros) should be located in the same folder. While there are multiple possible methods to extract metadata for each program, we used several straightforward SAS functions to obtain metadata on each SAS program.

Retrieve All SAS Program Names

In the example code below, the DOPEN function opens a directory and returns a directory identifier value that is used to identify the open directory in other SAS external file access functions. As our macro %PROCTOC specifies the macro parameter &TXTPATH as the directory of all SAS programs, DOPEN ('FOLDER') returns a value (which is greater than 0) and assigns it to the variable LIST. The following DO LOOP outputs each SAS file name to the temporary dataset FILESINFOLDER. In the DO LOOP section, DNUM is used to return the number of members in the SAS program folder and the DREAD function returns the name of a directory member and assigns it to the variable FILE.

```
/* Making a list of all txt files in the program folder */
filename folder "&TXTPATH";
data FILESINFOLDER;
    length LINE 8 FILE $300;
    LIST = dopen('folder');
```

```

do LINE = 1 to dnum(LIST);
    FILE = trim(dread(LIST,LINE));
    if index(File, '.txt') > 0 then output;
end;
drop LIST LINE;
run;

```

Retrieve Metadata for Each SAS Program

Given the possibility of hundreds of SAS programs, a DO LOOP iterating over all SAS files to obtain and output corresponding metadata is an attractive option. In order to utilize the DO LOOP to repeat the same programming assignment for various values of parameters, each SAS program's name needs to be coded into a macro variable and the total number of SAS programs needs to be counted and passed to the DO LOOP as well. As shown below, a simple data _NULL_ step is utilized where the CALL SYMPUT statement translates each SAS name into a dynamic macro variable and CALL SYMPUTX generates a dynamic macro variable 'TOTAL' for the count of all SAS files.

```

/* Creating global macro variables */
data _NULL_;
    set FILESINFOLDER end=final;
    call symput(cats('FILE', _N_), trim(FILE));
    if final then call symputx(trim('TOTAL'), _N_);
run;

```

Next, the %do i = 1 %to &TOTAL loop increments the value of i by one and sets "&TXTPATH\&&FILE&I" to a specific individual SAS program. Once the macro variables encoding each SAS program name are assigned and resolved, the next objective is to collect the metadata for each SAS file by using the FINFO function in conjunction with the FOPEN and FCLOSE functions, which are used to retrieve system information about each SAS file. To display all available attributes, we also use FOPTNUM and FOPTNAME where the FOPTNUM function returns the number of available information attributes for a specific SAS file and the FOPTNAME function returns the name of each of those information attributes. Figure 6 presents a simple example of available information attributes and their values retrieved from individual SAS files.

```

/*get metadata of each txt file*/
%macro loop;
data INFO;
    length INFONAME INFOVAL FILENAMES $600;
    %do i = 1 %to &TOTAL;
        rc=filename("TEMP", "&TXTPATH\&&FILE&I");
        FID=fopen("TEMP");
        INFONUM=foptnum(FID);
        do K=1 to INFONUM;
            FILENAMES= "&&FILE&I";
            INFONAME=foptname(FID,K);
            INFOVAL=finfo(FID, INFONAME);
            output;
        end;
        close=fclose(FID);
    %end;
run;
%mend loop;
%loop

```

FILENAMEX	INFONAME	INFOVAL
a_adae_sas.txt	Filename	C:\Study101\Analysis\ADaM\Programs\a_adae_sas.txt
a_adae_sas.txt	RECFM	V
a_adae_sas.txt	LRECL	32767
a_adae_sas.txt	File Size (bytes)	4086
a_adae_sas.txt	Last Modified	15Aug2018:08:21:49
a_adae_sas.txt	Create Time	26Mar2019:11:34:37
a_adbio_sas.txt	Filename	C:\Study101\Analysis\ADaM\Programs\a_adbio_sas.txt
a_adbio_sas.txt	RECFM	V
a_adbio_sas.txt	LRECL	32767
a_adbio_sas.txt	File Size (bytes)	5780
a_adbio_sas.txt	Last Modified	15Aug2018:08:21:43
a_adbio_sas.txt	Create Time	26Mar2019:11:34:37
a_adcm_sas.txt	Filename	C:\Study101\Analysis\ADaM\Programs\a_adcm_sas.txt
a_adcm_sas.txt	RECFM	V
a_adcm_sas.txt	LRECL	32767
a_adcm_sas.txt	File Size (bytes)	4033
a_adcm_sas.txt	Last Modified	15Aug2018:08:21:20
a_adcm_sas.txt	Create Time	26Mar2019:11:34:37
a_addf_sas.txt	Filename	C:\Study101\Analysis\ADaM\Programs\a_addf_sas.txt
a_addf_sas.txt	RECFM	V
a_addf_sas.txt	LRECL	32767
a_addf_sas.txt	File Size (bytes)	5199
a_addf_sas.txt	Last Modified	15Aug2018:08:27:22
a_addf_sas.txt	Create Time	26Mar2019:11:34:37

Figure 6: Metadata Retrieved for Each SAS Program

As demonstrated above, file attributes such as physical file location, file size, last modified time, etc. are extracted from each SAS file and set together in a temporary dataset called INFO. A simple PROC TRANSPOSE can convert the vertical structure to a horizontal TOC which displays file attributes per SAS file per row (Figure 7).

```
proc sort data=INFO;
    by FILENAMEX;
run;

proc transpose data=INFO out=SEC1(drop=_NAME_ RECFM LRECL);
    by FILENAMEX;
    id INFONAME;
    var INFOVAL;
run;
```


FILENAMES	FILENAME	FILE_SIZE_BYTES	LAST_MODIFIED	CREATE_TIME
a_adae_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_adae_sas.txt	4086	15Aug2018:08:21:49	26Mar2019:11:34:37
a_adbio_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_adbio_sas.txt	5780	15Aug2018:08:21:43	26Mar2019:11:34:37
a_adcm_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_adcm_sas.txt	4033	15Aug2018:08:21:20	26Mar2019:11:34:37
a_addf_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_addf_sas.txt	5199	15Aug2018:08:27:22	26Mar2019:11:34:37
a_adeg_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_adeg_sas.txt	4480	15Aug2018:08:30:03	26Mar2019:11:34:37
a_adiop_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_adiop_sas.txt	6178	15Aug2018:08:31:54	26Mar2019:11:34:37
a_adlb_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_adlb_sas.txt	5151	15Aug2018:09:29:30	26Mar2019:11:34:37
a_admh_sas.txt	C:\Study101\Analysis\ADaM\Programs\ad_admh_sas.txt	3669	15Aug2018:09:30:54	26Mar2019:11:34:37
align_sas.txt	C:\Study101\Analysis\ADaM\Programs\align_sas.txt	2077	01May2013:13:00:02	26Mar2019:11:34:37
filenum_sas.txt	C:\Study101\Analysis\ADaM\Programs\filenum_sas.txt	2430	11Jan2018:17:33:59	26Mar2019:11:34:37
formats_sas.txt	C:\Study101\Analysis\ADaM\Programs\formats_sas.txt	5380	15Aug2018:14:06:29	12Apr2019:10:17:54
initialize2_sas.txt	C:\Study101\Analysis\ADaM\Programs\initialize2_sas.txt	11915	04Jan2018:09:01:03	26Mar2019:11:34:37
lablngth_sas.txt	C:\Study101\Analysis\ADaM\Programs\lablngth_sas.txt	4872	16Jan2018:09:45:53	26Mar2019:11:34:37
stat_freq_sas.txt	C:\Study101\Analysis\ADaM\Programs\stat_freq_sas.txt	4047	10Jan2018:12:57:36	26Mar2019:11:34:37
stat_means_sas.txt	C:\Study101\Analysis\ADaM\Programs\stat_means_sas.txt	6384	22Apr2013:07:18:47	26Mar2019:11:34:37
stle_sas.txt	C:\Study101\Analysis\ADaM\Programs\stle_sas.txt	1701	07Jan2018:23:00:53	26Mar2019:11:34:37
t_ae1_sas.txt	C:\Study101\Analysis\ADaM\Programs\t_ae1_sas.txt	3100	16Aug2018:15:37:31	12Apr2019:10:17:55
t_ae2_sas.txt	C:\Study101\Analysis\ADaM\Programs\t_ae2_sas.txt	10926	15Aug2018:12:47:21	26Mar2019:11:34:37
t_ae3_sas.txt	C:\Study101\Analysis\ADaM\Programs\t_ae3_sas.txt	10455	15Aug2018:13:05:58	26Mar2019:11:34:37
t_aesev_sas.txt	C:\Study101\Analysis\ADaM\Programs\t_aesev_sas.txt	9860	15Aug2018:13:06:21	26Mar2019:11:34:37

Figure 7: Metadata Listed per Row per SAS Program

STEP 2: DETERMINE INPUT DATASETS AND GLOBAL MACROS USED

Multiple resources exist for extracting information on source datasets and global macro usage as well as associated output files for each submitted SAS program. One such resource is a study programming tracker, which initially may seem to be an easy option. A study programming tracker is a document used to track programming status for a deliverable, often implemented as an Excel file. It contains a list of all items required for a clinical study/project with multiple columns indicating corresponding SAS programs, their associated validation status, program creation time, last run time, and so on. Extra columns can also be added into a study programming tracker to include information on source datasets and global macro usage for each program and each unique output. One drawback of directly importing information from a study programming tracker into a TOC is the potential inconsistency between the information listed on the tracker and the actual source datasets and global macros used in the submitted program.

A second option is to list all source datasets and utilized global macros in a header for each SAS program and then extract them using keywords. However, this option also carries the possibility of inconsistency between the header and the actual program body.

Another option is to programmatically scan through the entire program body and extract source dataset(s) and utilized global macros. However, it is difficult to distinguish multi-line comments from actual programming codes and actual macro calls from free text.

Since the SAS log is a record of everything done in a SAS program, it might be the most reliable and straightforward way to extract source datasets and used global macros. Original program statements (including comments) are identified by line numbers while messages from SAS begin with the keywords NOTE, INFO, WARNING, or ERROR. Of particular interest, messages starting with NOTE are informational messages that indicate how SAS processes the program which contains information such as the source dataset name and the number of records inputted from an external file. Thus, we can easily distinguish the datasets imbedded in a multi-line comment versus actual usage of source datasets by using the keyword NOTE to identify the desired information (example code shown below).

In order to distinguish actual macro calls versus free text, we can easily extract log lines starting with the keyword MPRINT. MPRINT displays the SAS statements that are generated by macro execution, so a macro name following the keyword MPRINT is the actual macro call in the SAS program. Since internal macro code are already imbedded in the individual program, the list of macro names extracted using the

MPRINT keyword should be a subset of the user's central macro library. Only the utilized central macros are listed in the TOC (Figure 8).

```
/**since ADAM and TLF logs are stored in different folder pathway in our
organization, we created a macro %SCANLOG to scan through ADAM logs or TLF
logs - users can modify as needed **/
```

```
%macro SCANLOG (PATH=,TYPE=);
filename pfolder "&path.";
/**making a list of all log files in the folder**/
/**logic thoroughly explained in STEP 1 **/
Data LISTFILE;
    length LINE 8 FILE $300;
    LIST = dopen('pfolder');
    do LINE = 1 to dnum(LIST);
        FILE = trim(dread(LIST,LINE));
        FNAME=compress(tranwrd(FILE, '.log', ''));
        if index(FILE, '.log')>0 then output;
    end;
run;

/**get each log file into a macro variable**/
/**logic thoroughly explained in STEP 1 **/
data _null_;
    set LISTFILE end=eof;
    call symput("fname"||compress(put(_N_,8.)),compress(FNAME));
    if eof then call symput("COUNT", compress(put(_N_,8.)));
run;

/**scan each log file**/
/**DO LOOP logic thoroughly explained in STEP 1 **/
%do i=1 %to %eval(&COUNT);
    data INTEXT&I.;
        length TEXT FNAME $ 600 ;
        infile "&&PATH..\&&FNAME&I...LOG" end=eof truncover;
        input @1 TEXT $ 600. ;
        if (index(upcase(TEXT), 'ADATA.') and index(upcase(TEXT), 'NOTE:')) or
            (index(upcase(TEXT), 'SDTM.') and index(upcase(TEXT), 'NOTE:')) or
            index(upcase(TEXT), 'MPRINT') or
            index(upcase(TEXT), '%LET TABNUM');
        if index(upcase(TEXT), 'MPRINT') then
TEXT=scan(scan(TEXT,1,""),2,"(");
        ORD=_N_;
        FNAME="&&FNAME&I..";
        run;
    /**the complete code is not shown for length considerations of this paper **/
    ...
    ...
    ...
%end;

%mend;
```

FILENAMEX	OFNAME	DTUSED	MACUSED
a_adae_sas.txt	Create Dataset: ADAE	ADSL, AE	INITIALIZE2, LABLLNGTH
a_adbio_sas.txt	Create Dataset: ADBIO	ADSL, XO	INITIALIZE2, LABLLNGTH
a_adcm_sas.txt	Create Dataset: ADCM	ADSL, CM	INITIALIZE2, LABLLNGTH
t_mh_sas.txt	Create Table: 14.1.3.1	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT
t_mh_sas.txt	Create Table: 14.1.3.2	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT
t_urine_sas.txt	Create Table: 14.3.2.7	ADLB, ADSL	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT
t_va1_sas.txt	Create Table: 14.3.3.1	ADSL, ADVA	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT
t_va2_sas.txt	Create Table: 14.3.3.2	ADSL, ADVA	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT
t_va2_sas.txt	Create Table: 14.3.3.3	ADSL, ADVA	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT
t_vs_sas.txt	Create Table: 14.3.8.1	ADSL, ADVS	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT
t_vs_sas.txt	Create Table: 14.3.8.2	ADSL, ADVS	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT

Figure 8: Source Datasets, Global Macro Usage, and Corresponding Output for Each SAS Program

STEP 3: CREATE HYPERLINK TO ROUTE TOC TO PHYSICAL FILES

Tie Together Metadata and Program Input Information

A PROC SQL step shown below ties together retrieved metadata information on each SAS program with their dataset and any macro used and their produced output. Since FINFO function grabs and displays file size in bytes by default, we convert it to kilobytes for easy review. Now the merged dataset is one PROC REPORT away from the desired detailed TOC output (Figure 9).

```
proc sql;
    create table SEC3 as
    select a.*,b.*,strip(put(ceil(input(FILE_SIZE__BYTES_,8.)/1024),best.)) as
KB
    from SEC2 as a join SEC1 as b
    on a.FILENAMEX=b.FILENAMEX
    order by FILENAMEX, OFNAME;
quit;
```

FILENAMEX	OFNAME	DTUSED	MACUSED	KB	FILENAME
a_adae_sas.txt	Create Dataset: ADAE	ADSL, AE	INITIALIZE2, LABLLNGTH	4	C:\Study101\Analysis\ADaM\Programs\a_adae_sas.txt
a_adbio_sas.txt	Create Dataset: ADBIO	ADSL, XO	INITIALIZE2, LABLLNGTH	6	C:\Study101\Analysis\ADaM\Programs\a_adbio_sas.txt
a_adcm_sas.txt	Create Dataset: ADCM	ADSL, CM	INITIALIZE2, LABLLNGTH	4	C:\Study101\Analysis\ADaM\Programs\a_adcm_sas.txt
t_mh_sas.txt	Create Table: 14.1.3.1	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM\Programs\t_mh_sas.txt
t_mh_sas.txt	Create Table: 14.1.3.2	ADMH, ADSL	INITIALIZE2, FILENUM, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM\Programs\t_mh_sas.txt
t_urine_sas.txt	Create Table: 14.3.2.7	ADLB, ADSL	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM\Programs\t_urine_sas.txt
t_va1_sas.txt	Create Table: 14.3.3.1	ADSL, ADVA	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM\Programs\t_va1_sas.txt
t_va2_sas.txt	Create Table: 14.3.3.2	ADSL, ADVA	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM\Programs\t_va2_sas.txt
t_va2_sas.txt	Create Table: 14.3.3.3	ADSL, ADVA	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	6	C:\Study101\Analysis\ADaM\Programs\t_va2_sas.txt
t_vs_sas.txt	Create Table: 14.3.8.1	ADSL, ADVS	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM\Programs\t_vs_sas.txt
t_vs_sas.txt	Create Table: 14.3.8.2	ADSL, ADVS	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	7	C:\Study101\Analysis\ADaM\Programs\t_vs_sas.txt

Figure 9: Temporary Dataset Containing Program Metadata and Input Information

Hyperlink Creation

Hyperlinks or hypertext links provide direct access to a different section of the same document or provide direct access to a different folder or file location. Hypertext links that open a folder or document are set to open the linked folder or document in a new window. The output TOC with working hypertext links serves as a central location to open multiple related submission-based documents while displaying the related documents in one row.

While there are numerous techniques to build and maintain hyperlinks within SAS, we use the CALL DEFINE statement with the URL attribute in PROC REPORT to create a dynamic link based on a relative file to column values (example code shown below). First, we assign three temporary variables to store the

physical location of SAS programs: RPNAME, RDTNAME, and ROUTNAME. RPNAME presents the physical location of each submitted SAS program: table and figure programs, ADaM programs, or global macros that are in TXT format and placed in one folder per FDA eCDT requirement. RDTNAME points to an input dataset folder – either an ADaM or SDTM dataset folder where ADaM and SDTM are stored in XPT format. ROUTNAME points to the output location. In an e-submission, since the outputs for clinical study reports are not included in the folder structure defined by the eCTD technical conformance guide, we only link the output column to the ADaM or SDTM dataset folder. Users can update the code to link to any PDF outputs if needed. A relative path specifies the location of a file from the current directory downward and does not include the specific drive letter or parent directories above the current location. Using relative paths when creating hypertext links minimizes the loss of hyperlink functionality when submissions are loaded onto network servers.

After the physical paths are assigned and location variables are created, a compute block in PROC REPORT is utilized to initialize the CALL DEFINE (code shown below). The COMPUTE block establishes the variable to which we want to assign the link. In our case, they are FILENAMEX (Program Name), OFNAME (Output Name), DTUSED_WRP (Dataset Used), and MACUSED_WRP (Macro Used). Their corresponding variables are RPNAME, ROUTNAME, RDTNAME, and RMACNAME which store the physical location for individual files or folders. Using FILENAMEX and RPNAME as an example (Figure 10), the location stored in the variable RPNAME is a URL attribute value for FILENAMEX. This allows us to link individual cells of our TOC to another file or a specific folder. When creating links in tables of contents and throughout submission documents, we do not assign any border or color to minimize any confusion, but the users can adjust the style of the hyperlink in the 2nd CALL DEFINE statement with STYLE option.

The TOC with hyperlinks provides a convenient way to open and display a document as it exists (Figure 11). It enables reviewers to click and direct to a point of interest. The link will only be available if there is something for it to attach to in the cell. Since there is nothing assigned for TLF output in our example, there is no hyperlink for it. Nevertheless, code could be adapted to link each program with their corresponding PDF output.

*/**the complete code is not shown for length considerations of this paper **/*

```
data REPORT;
length RPNAME RDTNAME ROUTNAME $500;
set SEC3;
...
...
...
RPNAME("&TXTPATH."||strip(FILENAMEX));
RDTNAME("&ADOUT.");
RMACNAME("&TXTPATH.");
run;

option orientation=landscape;
ods listing close;
ods pdf file="&TXTPATH.\TOC.pdf" style=styles.pdftemp notoc ;
title4 J=C "Programs Submitted";
title5 j=c "Table of Contents";

proc report data=REPORT missing nowd headskip split="~" ;
column RPNAME FILENAMEX ...;
...
define RPNAME / noprint style(header)=[just=c] style(column)=[just=c
cellwidth=.4in ];
define FILENAMEX / "Program Name" style(header)=[just=l]
style(column)=[just=left cellwidth=1.3in asis=on];
...
...
```

```

compute  FILENNAMEX;
  call define (_col_, "URL", RPNAME );
  call define (_col_, 'STYLE', 'STYLE=[color=blue linkcolor=white] ');
endcomp;

compute  ofname;
  call define (_col_, "URL", ROUTNAME );
  call define (_col_, 'STYLE', 'STYLE=[color=blue linkcolor=white] ');
endcomp;

...
...
run;
ods pdf close;
ods listing;

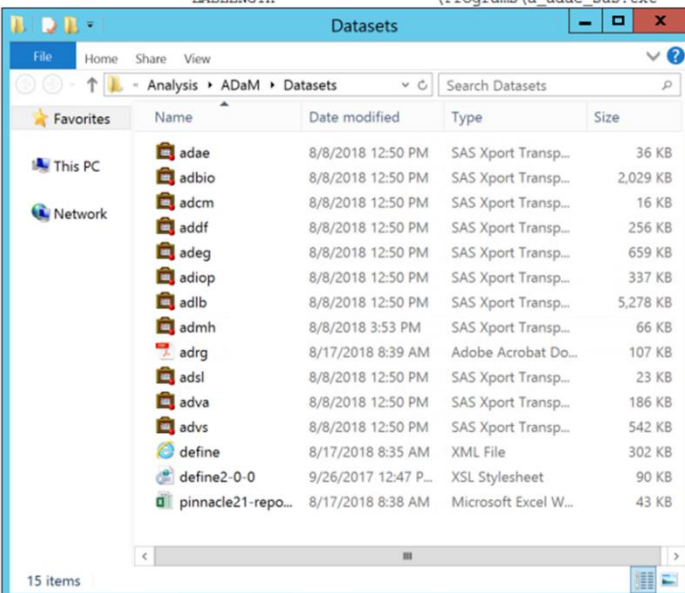
```

FILENNAMEX	RPNAME	OFNAME	ROUTNAME	DTUSED	RDTNAME	MACUSED	RMACNAME
a_adae_sas.txt	..\Programs\va_adae_sas.txt	Create Dataset: ADAE	..\Datasets	ADSL, AE	..\Datasets	INITIALIZE2, LABLLNGTH	..\Programs
a_adbio_sas.txt	..\Programs\va_adbio_sas.txt	Create Dataset: ADBIO	..\Datasets	ADSL, XO	..\Datasets	INITIALIZE2, LABLLNGTH	..\Programs
a_adcm_sas.txt	..\Programs\va_adcm_sas.txt	Create Dataset: ADCM	..\Datasets	ADSL, CM	..\Datasets	INITIALIZE2, LABLLNGTH	..\Programs
t_mh_sas.txt	..\Programs\vt_mh_sas.txt	Create Table: 14.1.3.1		ADMH, ADSL	..\Datasets	INITIALIZE2, FILENUM, STLE, TITFOOT	..\Programs
t_mh_sas.txt	..\Programs\vt_mh_sas.txt	Create Table: 14.1.3.2		ADMH, ADSL	..\Datasets	INITIALIZE2, FILENUM, STLE, TITFOOT	..\Programs
t_urine_sas.txt	..\Programs\vt_urine_sas.txt	Create Table: 14.3.2.7		ADLB, ADSL	..\Datasets	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT	..\Programs
t_va1_sas.txt	..\Programs\vt_va1_sas.txt	Create Table: 14.3.3.1		ADSL, ADVA	..\Datasets	INITIALIZE2, FILENUM, STAT_FREQ, STLE, TITFOOT	..\Programs
t_va2_sas.txt	..\Programs\vt_va2_sas.txt	Create Table: 14.3.3.2		ADSL, ADVA	..\Datasets	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	..\Programs
t_va2_sas.txt	..\Programs\vt_va2_sas.txt	Create Table: 14.3.3.3		ADSL, ADVA	..\Datasets	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	..\Programs
t_vs_sas.txt	..\Programs\vt_vs_sas.txt	Create Table: 14.3.8.1		ADSL, ADVS	..\Datasets	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	..\Programs
t_vs_sas.txt	..\Programs\vt_vs_sas.txt	Create Table: 14.3.8.2		ADSL, ADVS	..\Datasets	INITIALIZE2, ALIGN, FILENUM, STAT_MEANS, STLE, TITFOOT	..\Programs

Figure 10: Temporary Dataset Containing Program Metadata and Physical Location Information

Programs Submitted
Table of Contents

Program Name	Type	Class	Output Name	Input Dataset	Global Macro Used	Size (KB)	Location
a_adae_sas.txt	SAS	ADaM	Create Dataset: ADAE	ADSL, AE	INITIALIZE2, LABLLNGTH	4	C:\Study101\Analysis\ADaM\Programs\a_adae_sas.txt
a_adbio_sas.txt	SAS	ADaM	Create Dataset: ADBIO				
a_adcm_sas.txt	SAS	ADaM	Create Dataset: ADCM				
t_mh_sas.txt	SAS	Table	Create Table: 14.1.3.1				
t_mh_sas.txt	SAS	Table	Create Table: 14.1.3.2				
t_urine_sas.txt	SAS	Table	Create Table: 14.3.2.7				
t_val_sas.txt	SAS	Table	Create Table: 14.3.3.1				
t_va2_sas.txt	SAS	Table	Create Table: 14.3.3.2				
t_va2_sas.txt	SAS	Table	Create Table: 14.3.3.3				
t_vs_sas.txt	SAS	Table	Create Table: 14.3.8.1				



STAT_MEANS,
STLE,
TITFOOT

Figure 11: Hyperlinks in TOC Generated by %PROGTOC

CONCLUSION

The latest study data technical conformance guide specifically requires sponsors to include program code in the submission. A Table of Contents is expected to accompany these SAS programs for easy reference. In contrast to labor-intensive manual creation, this paper presents an efficient approach to extract metadata and input/output information of submitted programs to generate a detailed TOC report which helps enhance the quality of submission and enables the reviewer to navigate the document and access files easily. This automated process has great potential to help clinical trial sponsors save time and money while ensuring a high level of detail and accuracy.

REFERENCES

- US Food & Drug Administration. Oct. 2018. "Study Data Technical Conformance Guide V4.2. Accessed Nov. 15, 2018.
- US Food & Drug Administration. September 1998. "Guidance for Industry: E9 Statistical Principles for Clinical Trials". Accessed Dec. 15, 2018.
- US Food & Drug Administration. May 2007. "Guidance for Industry: Computerized Systems Used in Clinical Investigations". Accessed Dec. 12, 2018.
- US Food & Drug Administration. Jan. 2002. "General Principles of Software Validation; Final Guidance for Industry and FDA Staff". Accessed Dec. 15, 2018.
- US Food & Drug Administration. May 2015. "Statistical Software Clarifying Statement". Accessed Dec. 15, 2018.

US Food & Drug Administration. Nov. 2017. "Specifications for File Format Types Using eCTD Specifications." Accessed Dec. 15, 2018.

Maria Dalton 2019 "Submission of Software Programs to Regulatory Agencies" *Proceedings of PhUSE US Connect 2019*

Katrina E. Canonizado, Bradford J. Danner, and Matthew J. Wiedel. 2011. "A Non-Invasive Macro to Track Submission Metadata in SAS Drug Development" *Proceedings of PharmaSUG2011*.

Bradford J. Danner, Matthew J. Wiedel, and Katrina E. Canonizado. 2011. "Tracking Metadata within SAS Drug Development Using SDDPARMS" *Proceedings of PharmaSUG2011*.

Arthur L. Carpenter 2009 "Advanced PROC REPORT: Getting Your Tables Connected Using Links - Part I Tutorial" *Proceedings of SAS Global Forum2009*.

Jeff Xia, Lugang Xie (Larry) 2016 "A Well-Formatted and Easy-to-Navigate Solution for Submitting SAS Source Code in NDA Submission" *Proceeding of WUSS 2016*.

Angelo Tinazzi, Cedric Marchand 2017 "An FDA Submission Experience Using the CDISC Standards" *Proceeding of PhUSE 2017*

ACKNOWLEDGMENTS

The authors thank Kevin Uchimura, Kirk Bateman, Faith Kolb, and Melanie Ciotti at Statistics and Data Corporation for their thoughtful review of this manuscript.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Lingjiao Qi
Enterprise: Statistics & Data Corporation
Address: 63 South Rockford Drive, Suite 240, Tempe, AZ 85281
E-mail: lqi@sdcclinical.com
Web: <https://www.sdcclinical.com/>

Name: Bharath Donthi
Enterprise: Statistics & Data Corporation
Address: 63 South Rockford Drive, Suite 240, Tempe, AZ 85281
E-mail: bdonthi@sdcclinical.com
Web: <https://www.sdcclinical.com/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.