PharmaSUG 2015 - Paper QT44

Get a Data Dictionary using PROC contents: Easily

Beatriz Garcia, Inventiv Health Clinical Mexico, Mexico City

Alberto Hernandez, Inventiv Health Clinical Mexico, Mexico City

ABSTRACT

SAS programmers often have to work with a minimum set of documentation, and even the most experienced programmers find obstacles in the programming leading to the TLG's. In this example, we are given no variable names in the specifications (or also known as the spec file) or the annotated CRF is missing. A Data Dictionary would be a very good option for starting to analyze our data. This paper will present a macro with the use of PROC CONTENTS for creating a data dictionary that facilitates the way the knowledge of the project and can often reduce iterative questions to the Lead Programmer.

INTRODUCTION

When we start to work on our assignments, sometimes we do not receive the complete information and we want to analyze how the database is, but how we can do it if the documentation is minimum?

Besides that the programming is in UNIX environment in a remote session and the slowness is a big trouble.

Once we have reviewed the specifications, we notice that there are no variable names or the annotated CRF is missing, well, a Data Dictionary would be a very good option for starting to analyze our data.

In this paper we will introduce a macro that creates a very helpful Data Dictionary using PROC CONTENTS from a complete data library.

PROC CONTENTS

It Prints descriptions of the contents of one or more files from a SAS data library, it is often used by programmers for learning more about the details and attributes of a SAS dataset. When we run it into SAS, it looks like below:

```
                          CONTENTS PROCEDURE
Data Set Name: WORK.ORANGES                    Observations:          4
Member Type:   DATA                            Variables:             5
Engine:        V9                              Indexes:               0
Created:       15:56 Monday, April 27, 1999    Observation Length:   40
Last Modified: 15:56 Monday, April 27, 1999    Deleted Observations: 0
Protection:                                    Compressed:           NO
Data Set Type:                                 Sorted:              YES
Label:

                -----Engine/Host Dependent Information-----

       Data Set Page Size:         6144
       Number of Data Set Pages:   1
       First Data Page:            1
       Max Obs per Page:           152
       Obs in First Data Page:     4
       Number of Data Set Repairs: 0
       Physical Name:              SYS96050.T153830.RA000.USERID.R0000004
       Release Created:            8.0000B1
       Release Last Modified:      8.0000B1
       Created by:                 USERID
       Last Modified by:           USERID
```

```
Subextents:                 1
Total Blocks Used:          1
                Taste Test Results For Oranges

                     CONTENTS PROCEDURE

      -----Alphabetic List of Variables and Attributes-----

               #    Variable    Type    Len    Pos
               -----------------------------------
               2    FLAVOR      Num       8      8
               4    LOOKS       Num       8     24
               3    TEXTURE     Num       8     16
               5    TOTAL       Num       8     32
               1    VARIETY     Char      8      0
```

The procedure output provides values for the physical characteristics of the SAS data set WORK.ORANGES. Here we have the important values:

Observations	It is the number of nondeleted records in the data set.
Observation Length	It is the maximum record size in bytes.
Compressed	Has the value NO if records are not compressed; it has the value CHAR or BINARY if records are compressed.
Data Set Page Size	It is the size of pages in the data set.
Number of Data Set Pages	It is the total number of pages in the data set.
First Data Page	It is the number of the page that contains the first data record; header records are stored in front of data records.
Max Obs per Page	It is the maximum number of records a page can hold.
Obs in First Data Page	It is the number of data records in the first data page.

DATA DICTIONARY

A data dictionary contains a list of all datasets, descriptions of the variables, names and types of each ones.

It can be very valuable for programmers who need to understand and have a big picture of the data.

In the following macro, using Proc Contents we can get a complete data dictionary in few steps.

It needs of four macrovariables:

Libref: Define the library's name where we want to get the Data Dictionary.

Dset: Define the dataset's name, in this case we are using "_all_" since we want to get the definition for all datasets in the library.

Xlpath: Define the path where we want to save the Excel output.

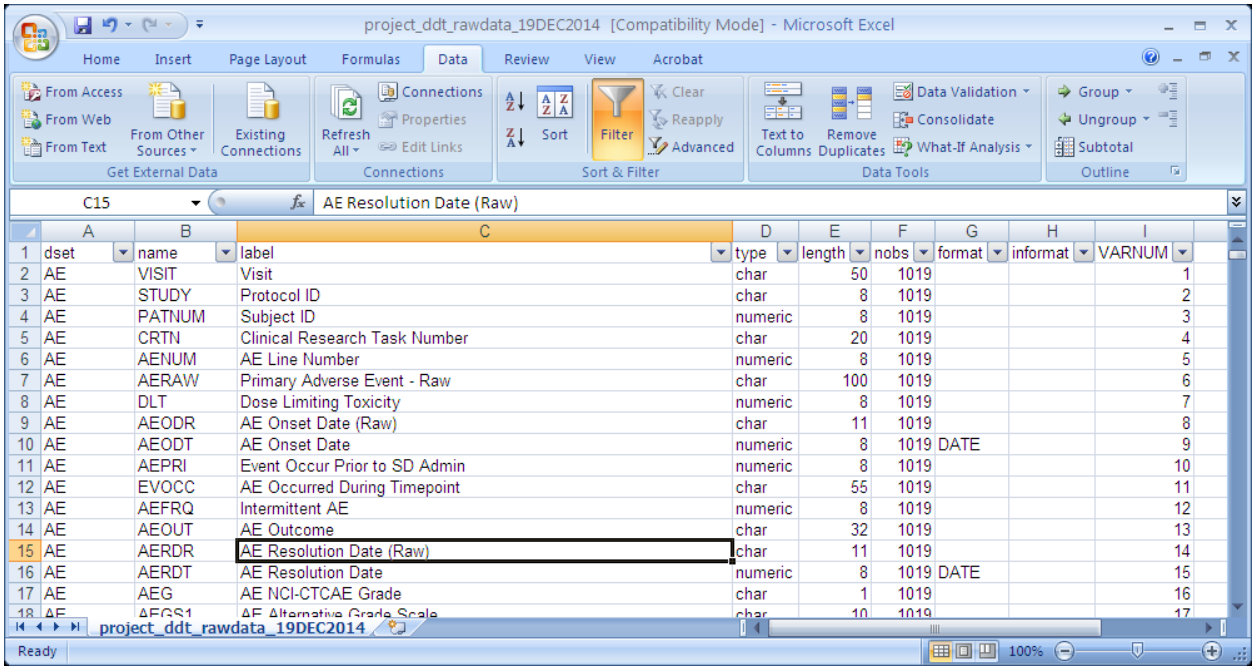
Xlname: Define the name of the Excel output, the macro adds the date to the output, using the macrovariable sysdate9.

We can define our library if it is required.

```
libname save  '/rawdata';

%macro data_dictionary(libref=work  , dset=_all_ , xlpath= , xlname= );
```

```
  proc contents noprint  data=&libref..&dset
       out=tmp_info(keep= memname name label type length varnum format informat nobs);
  run;

  data tmp_info;
    retain dset name label type length nobs format informat;
    set tmp_info (rename=(memname=dset type=typ));
        if typ=1 then type='numeric';
        else if typ=2 then type='char';
        drop typ;
  run;

  proc sort data=tmp_info;
    by dset varnum;
  run;

  %if &xlpath ne %str() %then %do;
    proc export  data=tmp_info
    outfile= "&xlpath./&xlname._&sysdate9..xls"
    dbms=xls
    replace;
    run;
  %end;


%mend data_dictionary;

%data_dictionary (libref=save,
                  xlpath=%str(/programs),
                  xlname=Project_ddt_rawdata
                  );
```

Once we run the code, we would get the following output:



	A	B	C	D	E	F	G	H	I
1	dset	name	label	type	length	nobs	format	informat	VARNUM
2	AE	VISIT	Visit	char	50	1019			1
3	AE	STUDY	Protocol ID	char	8	1019			2
4	AE	PATNUM	Subject ID	numeric	8	1019			3
5	AE	CRTN	Clinical Research Task Number	char	20	1019			4
6	AE	AENUM	AE Line Number	numeric	8	1019			5
7	AE	AERAW	Primary Adverse Event - Raw	char	100	1019			6
8	AE	DLT	Dose Limiting Toxicity	numeric	8	1019			7
9	AE	AEODR	AE Onset Date (Raw)	char	11	1019			8
10	AE	AEODT	AE Onset Date	numeric	8	1019	DATE		9
11	AE	AEPRI	Event Occur Prior to SD Admin	numeric	8	1019			10
12	AE	EVOCC	AE Occurred During Timepoint	char	55	1019			11
13	AE	AEFRQ	Intermittent AE	numeric	8	1019			12
14	AE	AEOUT	AE Outcome	char	32	1019			13
15	AE	AERDR	AE Resolution Date (Raw)	char	11	1019			14
16	AE	AERDT	AE Resolution Date	numeric	8	1019	DATE		15
17	AE	AEG	AE NCI-CTCAE Grade	char	1	1019			16
18	AE	AEGS1	AE Alternative Grade Scale	char	10	1019			17

SOME ADVANTAGES

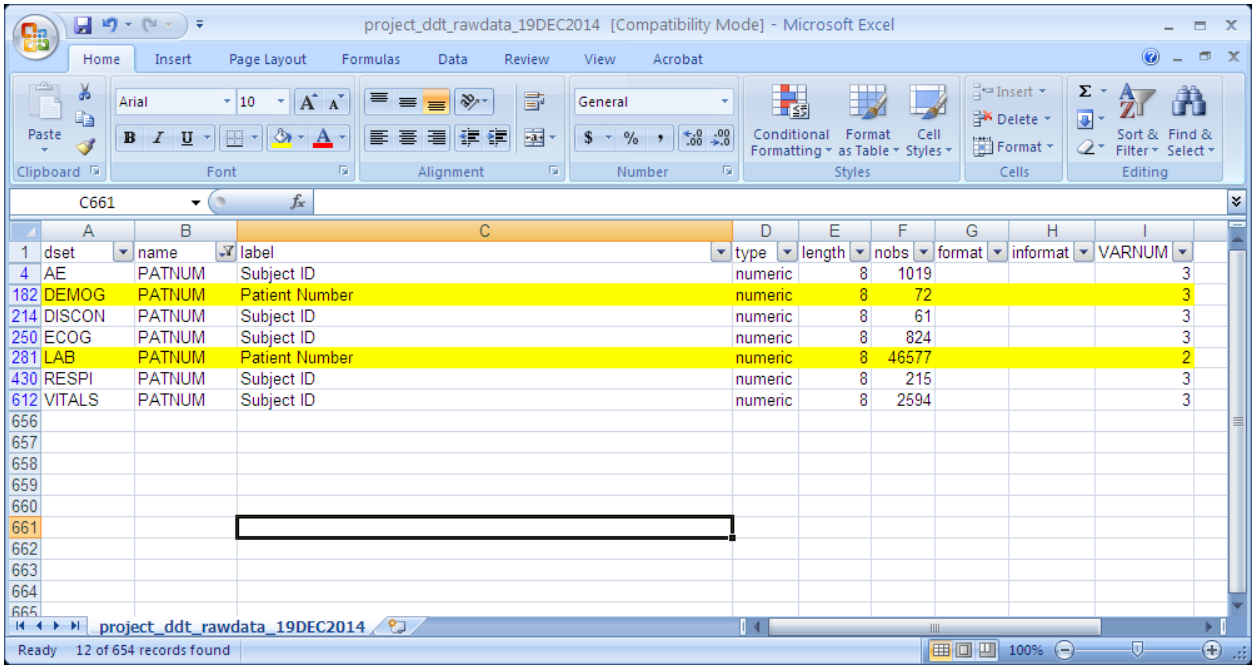
We can use all the functionality of Excel, i.e. in the figure below, using the filters we can review the following:

- Consistency at labels,
- Missing variables,
- Typos,
- Consistency at length or type.

This spreadsheet can be used as support for doing QC.

If you work using UNIX, you will find this spreadsheet very helpful and useful.

In the screenshot below, the highlighted rows show inconsistency in PATNUM variable.



	dset	name	label	type	length	nobs	format	informat	VARNUM
4	AE	PATNUM	Subject ID	numeric	8	1019			3
182	DEMOG	PATNUM	Patient Number	numeric	8	72			3
214	DISCON	PATNUM	Subject ID	numeric	8	61			3
250	ECOG	PATNUM	Subject ID	numeric	8	824			3
281	LAB	PATNUM	Patient Number	numeric	8	46577			2
430	RESPI	PATNUM	Subject ID	numeric	8	215			3
612	VITALS	PATNUM	Subject ID	numeric	8	2594			3

CONCLUSIONS

- With a Data dictionary we can save time when we are trying to understand how our database is built.
- When we do not have an accurate specs, we can use it in order to create it.
- We can use the data dictionary in Excel format for being sure which variables were created only using the filters.
- We can use this spreadsheet as spec in the creation of TLGs or Derived Datasets.

RECOMMENDED READING

- Base SAS® Procedures Guide
- http://support.sas.com/onlinedoc/912/docMainpage.jsp

ACKNOWLEDGEMENTS

Thanks to Mark Matthews for their support and guidance.

CONTACT INFORMATION

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

Name: Beatriz Garcia
Enterprise: Inventiv Health Mexico
Address: San Francisco #1005 Del Valle, Benito Juarez,
City, State ZIP: Mexico City, 03100
Work Phone: +52 55 5827 0917
E-mail: beatriz.garcia@inventivhealth.com

Name: Alberto Hernandez
Enterprise: Inventiv Health Mexico
Address: San Francisco #1005 Del Valle, Benito Juarez,
City, State ZIP: Mexico City, 03100
Work Phone: +52 55 5005 5507
E-mail: alberto.hernandez@inventivhealth.com