

Paper 236-2011

Using Microsoft Access 2007 Description of Field Names in Tables as Variable Labels in SAS® Data Sets

Maria S. Melguizo Castro, Jerry R Stalnaker, and Christopher J. Swearingen
Biostatistics Program, Department of Pediatrics
University of Arkansas for Medical Sciences, Little Rock, AR

ABSTRACT

It is a common practice for statisticians and data analysts to import data from Microsoft Access into SAS for analysis. Although it is relatively straightforward to transfer data between the two programs, currently there is not an option in SAS to import the description of field names from Access tables. Any good SAS programmer would be interested in using those descriptions as labels for variables in SAS data sets. This paper proposes a solution using Visual Basic for Applications (VBA) and SAS macros, first to generate a table of field names, descriptions, and table names in Access; and second, to use this table to create the labels in the SAS data sets.

INTRODUCTION

The problem arises when statisticians want to use Microsoft Access 2007® description of field names in the database tables as labels for SAS dataset variables. SAS v9.2 does not have the capability of importing descriptions from Access. Also, Microsoft Access does not allow the analyst to copy this information, available from the table in the design view, in a simple manner as each description has to be copied manually field by field. This process can be really cumbersome, especially for large databases.

A Visual Basic for Applications (VBA) macro using Microsoft Access 2007® was created to generate a metadata table, consisting of field names, descriptions, and table names in the database. The new table is used to create labels for variables in SAS datasets. The process is illustrated with an example database.

ACCESS DATABASE

An example of a MS Access database is presented below to mimic a data collection instrument typical of a research study. The database contains 3 tables: Demographics, Test and Questionnaire. **Figure 1** presents the relationships between the tables, and **Figure 2** shows the tables in Design View.

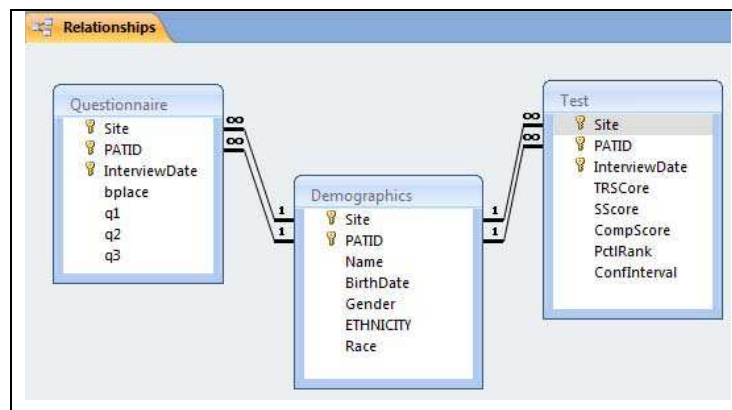



Figure 1. Database and Relationships

Using Microsoft Access 2007 Description of Field Names in Tables as Variable Labels in SAS® Data Sets, continued

Demographics			Test			Questionnaire		
Field Name	Data Type	Description	Field Name	Data Type	Description	Field Name	Data Type	Description
Site	Number		Site	Number	Study Site	Site	Number	Study Site
PATID	Number	Participant ID	PATID	Number	Participant ID	PATID	Number	Participant ID
Name	Text	Participant's Name	InterviewDate	Date/Time	Date of the Interview	InterviewDate	Date/Time	Date of the Interview
BirthDate	Text	Date of Birth	TRScore	Number	Total Raw Score	bplace	Text	Place of Birth
Gender	Text	Participant's Gender	SScore	Number	Scaled Score	q1	Number	Question 1
ETHNICITY	Text		CompScore	Number	Composite Score	q2	Number	Question 2
Race	Text	Participant's Race	PctRank	Number	Percentile Rank	q3	Number	Question 3
			ConfInterval	Number	Confidence Interval			

Figure 2. Tables in Design View

ACCESS VBA AND SAS CODE

In the Microsoft Visual Basic window, a new Module  Module was inserted to create a VBA macro. The following code will create a new table in the Access database that contains all of the field names, descriptions and names for all the tables in the database. This code should be run first before invoking the SAS code. The macro to be run is *CreateLabels()*; the *CreateTable()* code is a sub routine that creates the blank scheme of the table Descriptions. This code can be used in any database; the only concern will be a database that contains a table called Descriptions (because this VBA macro would delete such table); in this case, the analyst may need to rename such table.

VBA CODE

```
' Auxiliary sub routine CreateTable

Sub CreateTable()
    Dim wspDefault As Workspace, dbs As Database
    Dim tdf As TableDef, fld1 As Field, fld2 As Field
    Dim idx As Index, fldIndex As Field

    Set wspDefault = DBEngine.Workspaces(0)
    ' Open Current database.
    Set dbs = CurrentDb()

    ' Create new table with three fields.
    Set tdf = dbs.CreateTableDef("Descriptions")
    Set fld1 = tdf.CreateField("VariableName", dbText)
    Set fld2 = tdf.CreateField("VariableDescription", dbText, 50)
    Set fld3 = tdf.CreateField("tblNAME", dbText)

    ' Append fields.
    tdf.Fields.Append fld1
    tdf.Fields.Append fld2
    tdf.Fields.Append fld3

    ' Append TableDef object.
    dbs.TableDefs.Append tdf
    dbs.TableDefs.Refresh
    Set dbs = Nothing

End Sub
```

Using Microsoft Access 2007 Description of Field Names in Tables as Variable Labels in SAS® Data Sets, continued

```
' Main VBA subroutine
Sub CreateLabels()
    Dim tdf As TableDef
    Dim db As Database
    Dim fld As Field
    Dim prp As Property
    Dim pos As Integer
    Dim repl, replwith As String
    Dim repldesc, repldescwith As String
    Dim tname As String
    Dim rst As Recordset
    Dim i As Integer
    Dim SQL As String
    Dim DescrAvailable As Boolean

    Set db = CurrentDb()

    'If the table descriptions exist then delete it

    For i = 0 To db.TableDefs.Count - 1
        If db.TableDefs(i).Name = "Descriptions" Then
            db.TableDefs.Delete "Descriptions"
            db.TableDefs.Refresh
            Exit For
        End If
    Next

    'the macro CreateTable will create a table with the Descriptions
    Call CreateTable
    Set rst = db.OpenRecordset("Descriptions")

    For Each tdf In db.TableDefs
        For Each fld In tdf.Fields
            DescrAvailable = False
            For Each prp In fld.Properties
                If prp.Name = "Description" Then
                    With rst
                        .AddNew
                        !Variablename = fld.Name
                        !VariableDescription = prp.Value
                        !tblNAME = tdf.Name
                        .Update
                        DescrAvailable = True
                    End With
                End If
            Next
            If DescrAvailable = False Then
                With rst
                    .AddNew
                    !Variablename = fld.Name
                    !tblNAME = tdf.Name
                    .Update
                End With
            End If
            DescrAvailable = False
        Next
    Next
    db.Close
End Sub
```

SAS V9.2 CODE

The first part of the SAS code was created to import the three data tables available in the database, as well as the Description table. The second part of the code takes the Description table and uses it to create labels into the three newly created SAS data sets.

To make this code useful for any specific data analysis, the analyst would need to modify the database path, and create as many *%ImportTables()*, and *%Labels()* macro invocations as tables in the database.

```

/*****
/* Import access tables*/
*****/

%Macro ImportTables(TableName);
PROC IMPORT OUT= WORK.&TableName
           DATATABLE= "&TableName"
           DBMS=ACCESS REPLACE;
           DATABASE="C:\SUGI\Databases\SUGI.accdb";
           SCANMEMO=YES;
           USEDATE=NO;
           SCANTIME=YES;
RUN;
%mend;

%ImportTables(Demographics);
%ImportTables(Test);
%ImportTables(Questionnaire);
%ImportTables(Descriptions);

/*****
/* Macro to Create Labels using table Descriptions */
*****/

%macro Labels(TableName);
proc sql;
  select count(*) into:numvar from Descriptions where
tblNAME="&TableName";
  %let numvar =&numvar;
  select VariableDescription into :varlab1-:varlab&numvar from
Descriptions where tblNAME="&TableName";
  select VariableName into :varname1-:varname&numvar from
Descriptions where tblNAME="&TableName";

  Data &TableName;
    set &TableName;
    %do i=1 %to &numvar;
      label &&varname&i="&&varlab&i";
    %end;
run;quit;
%mend;

%Labels(Demographics);
%Labels(Test);
%Labels(Questionnaire);

```

Using Microsoft Access 2007 Description of Field Names in Tables as Variable Labels in SAS® Data Sets, continued

CONCLUSION

It is a common practice between statisticians and data managers to import data collected in Access into SAS for analysis. The current paper shows that the capabilities of SAS can be expanded by the use of VBA macros.

REFERENCES

- ◆ Roper, Christopher A. *Using SAS and DDE to execute VBA macros in Microsoft Excel*. Qualex Consulting Services, Inc., Cary, NC. USA. Available at <http://www2.sas.com/proceedings/sugi25/25/cc/25p098.pdf>
- ◆ Lu, Zaizai; Shen, David. *Data Transfer from Microsoft Access to SAS Made Easy*. AstraZeneca Pharmaceutical and, ClinForce Inc. Available at <http://www.lexjansen.com/pharmasug/2005/coderscorner/cc12.pdf>
- ◆ Bessler, LeRoy. *SAS®-with-Excel Application Development: Tools and Techniques.*, Assurant Health, Milwaukee, Wisconsin, USA. Available at <http://www2.sas.com/proceedings/sugi31/022-31.pdf>
- ◆ Nan, Rubin; Mullins, David. *A Microsoft® Access GUI for SAS® Automation*. PRA International, Lenexa, KS. Available at <http://www2.sas.com/proceedings/sugi30/044-30.pdf>
- ◆ Luo, Lian; Yiin, James Ph.D. *A VB.NET utility to transfer multiple data files between SAS® and Microsoft® Access/Excel by automatically generating SAS® Import/Export statements*. SRA International Inc, Cincinnati, OH. National Institute for Occupational Safety and Health, Cincinnati, OH. Available at <http://support.sas.com/resources/papers/proceedings10/191-2010.pdf>
- ◆ Shvorob, Dimitri. *I Want It All - Recovering Variable Labels When Importing Data from Microsoft Access*. Vanderbilt University, Nashville, TN. Available at: <http://www2.sas.com/proceedings/forum2007/003-2007.pdf>

CONTACT INFORMATION

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

Name: Maria S. Melguizo Castro
Enterprise: University of Arkansas For Medical Sciences UAMS - Department of Pediatrics
Address: 1 Childrens Way, Slot 512-43
City, State ZIP: Little Rock, AR 72202-3500
Work Phone: (501) 364-6619
Fax: (501) 364-1431
E-mail: msmelguizo@uams.edu
Web: <http://www.arpediatrics.org/research/biostatistics>

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.