

Paper CT-25

A Handy New SAS® Tool for Comparing Dynamic Datasets

LaTonia Richardson, Centers for Disease Control and Prevention, Atlanta, GA

ABSTRACT

A new data comparison tool provides an efficient way for users to compare two dynamic datasets created at different times. The tool can compare datasets of various formats, including SAS, Access, Excel, and XML. Using a Microsoft Access “front-end” user interface to execute SAS Data Step and Proc SQL code, users are prompted to import two files of the same file format (an “old file” and a “new file”) and then thorough comparisons are made to identify differences in variables, records (i.e., records found in only one dataset), and data values. Once all comparisons are made, users can view user-friendly summary reports of all differences identified by the tool. The tool is most useful for tracking changes to dynamic datasets and for detecting, investigating, and resolving data discrepancies. It provides a convenient, efficient method for identifying key differences between two datasets. This paper reviews the features of this new tool, common uses for it, and the programming techniques used to create it.

INTRODUCTION

Working with dynamic datasets can be challenging. Differences between consecutive data downloads often require manual review in order to pinpoint data discrepancies. Manually detecting data value differences is reasonable for small datasets with few variables but is nearly impossible for large datasets with thousands of records and many variables. This paper presents a new tool that provides an efficient way to detect differences in two dynamic data sets captured at different points in time.

PREREQUISITES

Prior to running the data comparison tool, there are certain preliminary requirements that the imported datasets must satisfy. First, the file formats must be the same in order to eliminate false discrepancies that may occur due to differences in how the files are imported. Thus, an Access file can only be compared to another Access file and not to an Excel file or a SAS dataset. Next, the datasets must contain a common variable that uniquely identifies each record (a primary key), and there must be at least one record common to both datasets (i.e., the same primary key value in both datasets). Finally, for Access files only, the user and group permissions must be set to ‘Administer’ for the ‘MSysObjects’ and ‘MSysRelationships’ system tables. Here are the steps for setting the user and group permissions within an Access database:

1. Right-click the ‘Tables’ panel and select ‘Navigation Options’ (see Figure 1 below):

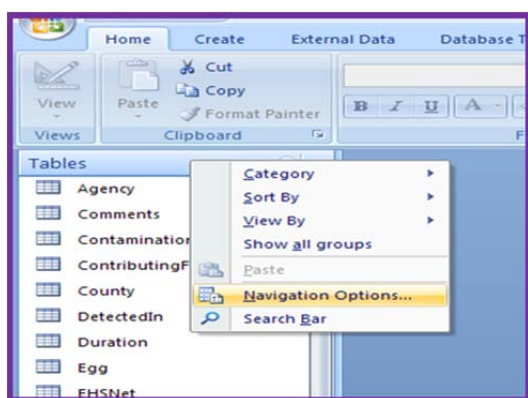


Figure 1. Navigation Options Menu in Access 2002-2003

2. Under ‘Display Options’ make sure that the ‘Show System Objects’ box is checked.
3. Go to the ‘Database Tools’ Menu and select ‘Users and Permissions’. Under ‘Object Name’ select the ‘MSysObjects’.

4. In the 'List' section, 'Users' should be selected by default. Check the 'Administer' box under 'Permissions' (this will automatically check all other check boxes). Click 'Apply' (see Figure 2 below):

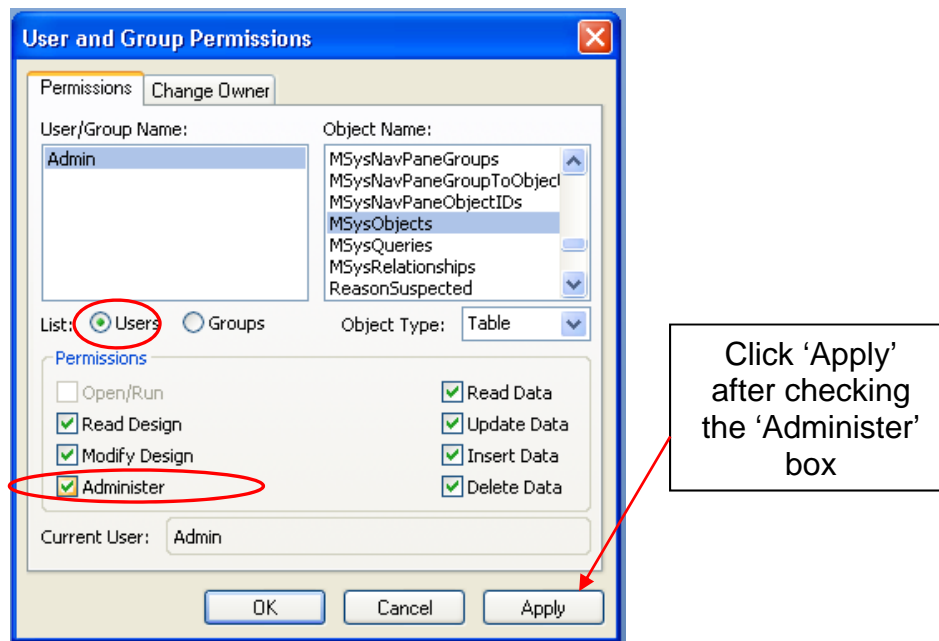


Figure 2. Setting User Permissions in Access 2002-2003

5. Next, select 'Groups' in the 'List' area. Under User/Group Name, 'Admin' should be selected by default. Under 'Permissions' select 'Administer' (this will automatically check all other check boxes). Then, click 'Apply' (see Figure 3 below):

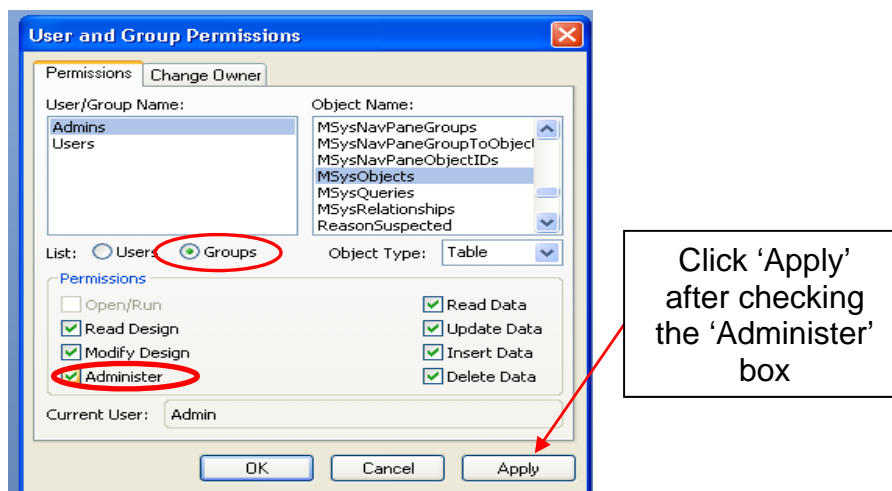


Figure 3. Setting Group Permissions in Access 2002-2003

6. Next, under User/Group Name, select 'User'. Under 'Permissions' select 'Administer' (this will automatically check all other check boxes). Then, click 'Apply'
7. Repeat Steps 3-6 for the 'MSysRelationships' Object. Once completed, click 'OK' to exit the "User and Group Permissions" window.

RUNNING THE TOOL

The data comparison tool uses a Microsoft Access front-end user interface with user-friendly menu options for easy navigation (see Figure 4). To run the tool, users must first enter the name of the primary key variable common to both datasets. Then, users must click the "Click Here To Run Comparisons" button, which will prompt them to select the two datasets that they wish to compare. The tool can accommodate datasets of various formats, including SAS, Access, and Excel. By default, a relational file structure is assumed for Access and Excel files (i.e., multiple tables and worksheets, respectively), but flat structures are also accommodated. It is important to select the same file type for both datasets (e.g., two Excel files, two Access files, etc.) in order to ensure an analogous comparison. Also, since the comparison tool is designed to detect changes in data files over time, the tool works best if one data file represents a "newer" version of the other data file. After selecting the two files to be compared, a pop-up warning is displayed which cautions the user that the tool may run for several minutes depending on the size of the datasets. The tool may run for well over 30 minutes when there are a large number of comparisons being made across very large datasets. A confirmatory message appears at the end of the process to notify users when the process is completed, allowing users to minimize the tool and run it in the background as they await confirmation.

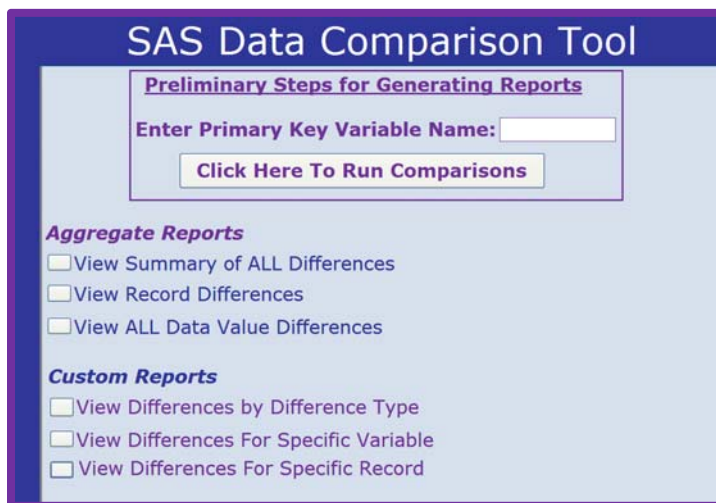


Figure 4. SAS Data Comparison Tool User Interface

SAS CODE

The key to the comparison tool's functionality is that it interfaces Microsoft Visual Basic programming language with SAS macro language to enable Access and SAS to communicate effectively with one another. For example, after the user enters the name of the primary key variable in the Access interface, this value is stored as a temporary Visual Basic variable. Using this Visual Basic variable, Access "calls" SAS and passes the variable over as a SAS macro variable. Subsequent programming codes are transferred from Access to SAS in a similar manner.

Once SAS is initiated from the Visual Basic environment, it performs all of the work for comparing the imported datasets. The first step is to determine the file type of the imported files. Once the user selects the two files within Access, the name of the files are passed over to SAS as macro variables called "dbpath1" and "dbpath2". SAS determines the file type by extracting the file extension from the macro variable names. For example, if the file extension is ".mdb" or ".accb", then SAS determines that the files are Access files (please see sample code below). Determining the file type helps to determine whether a relational or flat data structure should be considered.

```
%let dbpath1=C:\My Documents\MyFile1.mdb;
%let dbpath2=C:\My Documents\MyFile2.mdb;

data temp;
  fndAccess1=index(upcase("&dbpath1."), ".MDB");
  fndAccess2=index(upcase("&dbpath1."), ".ACC");
  fndExcel=index(upcase("&dbpath1."), ".XLS");
  fndSAS=index(upcase("&dbpath1."), ".SAS");
  length filetype $50.;
  if (fndAccess1>0 or fndAccess2>0) then filetype="ACCESS";
```

```

if (fndExcel>0) then filetype="EXCEL";
if (fndSAS>0) then filetype="SAS";
run;

```

After determining the file type of the comparison data sets, SAS first compares the variables to see if a particular variable is found in only one dataset. To do this, it runs the PROC CONTENTS command to generate a list of variables for each dataset. Then, it merges the two variable lists and outputs the variables that are found in only one dataset.

Next, for Access and Excel files only, SAS compares the tables (or worksheets) to see if there are tables found in only one dataset. For example, imagine an Access database with individual tables corresponding to each U.S. State. If this database is compared to another database with tables corresponding to both U.S. States and territories, SAS will flag the territory tables as discrepancies since they were not found in the first database.

For Access files only, SAS also checks for discrepancies between variable relationships set the by user. This comparison uses the 'MSysRelationships' table described in the 'Prerequisites' section above.

For all file types, SAS also compares the records in both datasets by using the primary key variable specified by the user. If a primary key variable is found in only one file, then SAS flags the record as a discrepancy.

Finally, SAS checks for data value differences between variables common to both datasets. This is the most thorough comparison performed and can also be the most time-consuming depending on the number of variables. Using an iterative "do loop" procedure, SAS compares the "old" and "new" values of each variable in every record and then categorizes the differences into one of three groups: 1) Missing to Non-Missing, which refers to records with a missing value in the "old" dataset and a non-missing value in the "new" dataset ; 2) Non-Missing to Missing, which refers to records with a non-missing value in the "old" dataset and a missing value in the "new" dataset ; and 3) Changed Value, which refers to records with discrepant non-missing values in the "old" and "new" datasets.

After making all of the comparisons, SAS exports all discrepancies to an Access database. The tables in this database are automatically linked to the initial Access user interface to populate the summary reports.

SUMMARY REPORTS

Data discrepancy summary reports are an additional feature of the SAS data comparison tool (see Figure 4 above). Reports are separated into two types: aggregate reports and custom reports. A description of each summary report is presented below:

Aggregate Reports

- View Summary of ALL Differences—Allows the user to view the results of all data discrepancies, including Table Name, Variable Name, Records, Relationships, and Data Value differences.
- View Record Differences--Allows the user to view the results of record discrepancies; provides a list of the records found in only one dataset
- View ALL Data Value Differences---Allows the user to view all data value differences. The variables and records common to both datasets were used to make the comparisons. For each record, the user may view the "old value" and "new value" for a given variable.

Custom Reports

- View Detailed Data Value Differences by Difference Type—Allows the user to filter data value differences by difference type, where difference type may be one of the following: Missing to Non-Missing, Non-Missing to Missing, or Changed Value.
- View Detailed Data Value Differences For Specific Variable---Allows the user to view data value differences for a specific variable.
- View Detailed Data Value Differences For Specific Record-----Allows the user to view data value differences for a specific record.

CONCLUSION

The SAS data comparison tool provides a more efficient method for identifying data discrepancies between dynamic datasets. Using a point-and-click interface to execute SAS commands, a novice SAS user with no programming expertise can easily run the tool. Although the tool is very thorough in its analysis, it is important to note that the single most accurate method for identifying differences between two data sets is with the human eye. However, the practical limitation of viewing several thousands of records manually makes this unrealistic. Thus, users should acknowledge that this tool provides an efficient way to identify all *potential* differences between data sets, but final confirmation of *actual* differences should be made by verifying the summary results generated by the tool.

CONTACT INFORMATION

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

LaTonia Richardson, MS
Statistician
Centers for Disease Control and Prevention
1600 Clifton Road, MS C-09
Atlanta, GA 30333
LCRichardson@cdc.gov

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.