

Let SAS “Modify” Your Excel File

Nelson Lee, Genentech, South San Francisco, CA

ABSTRACT

It is common to export SAS data to Excel by creating a new Excel file. However, there are times that we want to update the input Excel file instead of creating a new one so that we can preserve certain data attributes such as text highlight, font color, and etc. This paper shares a quick tip that utilizes the SAS MODIFY statement to update a column in an Excel file where the input file and the output file are the same. The advantage of this approach, documented in this paper, is that it reads data from an edit check specifications document (an Excel file) and it updates a targeted column in the same document.

This paper is written for audiences with beginner skills; the code is written using SAS Version 9.2 on the Windows operating system.

INTRODUCTION

In Clinical Data Management (CDM), one of the major goals is to ensure the data quality in conducting clinical trials. A common practice in the industry is to implement edit checks. While there are different ways to document edit checks, edit checks documented in an Excel spreadsheet is one of the most popular options. This document is commonly referred to as the Edit Check Specifications (ECS) document.

Once the ECS is developed, Clinical Programmers (CP) will start edit check programming based on the ECS and track programming progress on the same specification during the programming cycle. It is labor intensive to manually track if a check has been programmed or not, and thus SAS is used to facilitate this tracking process by comparing the ECS with other specifications. For example, one can generate a report from an Electronic Data Capture (EDC) system that holds the relevant edit check information such as check names. The comparison concept is fairly straightforward; it compares the data from the ECS with the data from the EDC system, identifies matching check names and then generates a report (a new document) to indicate if a check has been programmed or not.

One challenge is that the ECS is a living document. Data Managers (DM) and Clinical Programmers will need to modify it from time to time. It is challenging to maintain track changes in the ECS because Excel does not have the same level of track changes functionality as in Microsoft Word. Color fonts, highlights, and strikethroughs are some Excel attributes that are utilized for the purpose of tracking or identifying changes. For example, a check description highlighted in yellow could mean this check has been modified or a strikethrough on the check message could mean that certain part of the message is no longer needed. Hence, there is a need to maintain these attributes.

It is difficult to track progress and changes in two different documents when there is no automatic direct link between them. This paper shares a different approach by modifying/updating the ECS to maintain tracking progress while keeping certain ECS (Excel) attributes as described above and avoiding the creation of another report. In other words, the input file and the output are the same file (i.e., ECS).

AN EXAMPLE

Exhibit 1 below illustrates an edit check specification in Excel format. It has highlights, color fonts and strikethroughs for predefined reasons. Each of these attributes stand for something that is meaningful to the CP and DM. In this example, there are 2000 edit checks in the ECS. It is easy to imagine how labor intensive it will be to track programming progress by manually updating the Programmed column to indicate when a check is programmed. Needless to say, it is also prone to human error.

Exhibit 2 is a Check Listing (CL), an extract from an EDC system that documents all checks that exist in the system. These checks have been programmed as specified in the ECS. In this example, the CL is also an Excel file and has 500 checks.

Records from the ECS and the CL (input files) are read into SAS and compared by check name. The programming logic is to identify matching check names between the 2 documents and to populate the Programmed column in the ECS with “Yes” or “No”. “Yes” means that the same check name exists in the EDC system (i.e., the edit check has been programmed). This aims to eliminate the effort that manual tracking entails and to avoid human entry errors.

ID	Programmed	eCRF	Check description	Check Message	Action Field	Check Name
1		Demographics	Subject age must be at least 18 years old at screening	Subject is younger than 18 years old. Please review entries.	AGE	age_less_than_18
2		Demographics	Gender must be provided.	A response is required.	SEX	sex_missing
3		Lesions	Lesion number # do not match previous visit	Lesion # at this visit not matching the previous visit. Please review entries.	LESION_NO	lesion_no_not_match_previous_visit
4		Tumor Assessment	Assessment is marked Yes on the assessment form, Tumor response form must be submitted.	Assessment is answered 'Yes' but there is no matching Tumor Response submitted. Please review entries.	ASSESSMENT	assessment_yes_no_tumor_response form
5		Chemistry	If Not Done is not reported then Collection Date is required. This check is optional.	Collection date is required.	COLL_DATE	lab_col_date_missing
...	
2000		Vital Signs	Vital sign date must be on or before study completion date.	Vital signs date is after the study complete date. Please review entries.	VS_DATE	vitalsigns_date_after_completion

Exhibit 2 (Check Listing)

Record	Check Name
1	age_less_than_18
2	dbp_too_high
3	assessment_yes_no_tumor_response form
...	...
500	vitalsigns_date_after_completion

CREATE A NEW REPORT

The following code compares the two input files and assigns "Yes" to the variable 'Programmed' if the check name matches. This indicates that the edit check is programmed as it exists in the EDC system. A new report named 'outfile.xls' is generated (refer to Exhibit 3). Note that Exhibit 3 captures the correct information and resembles the original edit check specification with updated data in the Programmed column. However, the color fonts or highlighted attributes are no longer there.

```

*prepare libname to read input files;
libname ecs "C:\Users\nlee01\Desktop\MY SAS\editcheck.xls" ;
libname cl "C:\Users\nlee01\Desktop\MY SAS\checkedcdc.xls" ;

*create data set for edit checks from edit check specification;
data checkspec;
    set ecs.'editcheck$'n;
run;

*create data set for checks programmed in EDC system;
data checklist;
    set cl.'checkedcdc$'n;
run;

*prepare data sets for merging;
proc sort data=checkspec; by check_name; run;
proc sort data=checklist; by check_name; run;

*merge data and identify matching check name;
data matchcheck;
    merge checkspec(in=in1) checklist(in=in2);
    by check_name;
    if in1;
    if in1=in2 then programmed='Yes';
    else programmed ='No';
run;

proc sort; by id; run;
    
```

```

*write output to Excel;
ods listing close;
ods tagsets.excelxp file="C:\Users\Nlee01\Desktop\MY SAS\outfile.xls";

ods tagsets.excelxp options(sheet_name='editcheck' Absolute_Column_Width="5,8,5,30,20,10,20"
                             autofit_height = 'yes' autofilter ='all');

proc print data=match | noobs;
var ID Programmed eCRF Check_Description Check_Message Action_Field Check_Name;
  label
    Check_Description = 'Check Description'
    Check_Message = 'Check Message'
    Action_Field = 'Action Field'
    Check_Name = 'CheckName'
  ;
run;

ods tagsets.excelxp close;
ods listing;

libname ecs clear;
libname cl clear;

```

Exhibit 3

ID	Programmed	eCRF	Check Description	Check Message	Action Field	CheckName
1	Yes	Demographics	Subject age must be at least 18 years old at screening	Subject is younger than 18 years old. Please review entries.	AGE	age_less_than_18
2	No	Demographics	Gender must be provided.	A response is required.	SEX	sex_missing
3	No	Lesions	Lesion # do not match previous visit	Lesion # at this visit not matching the previous visit. Please review entries.	LESION_NO	lesion_no_not_match_previous_visit
4	Yes	Tumor Assessment	Assessment is marked Yes on the assessment form, Tumor response form must be submitted.	Assessment is answered 'Yes' but there is no matching Tumor Response submitted. Please review entries.	ASSESSMENT	assessment_yes_no_tumor_response form
5	No	Chemistry	If Not Done is not reported then Collection Date is required. This check is optional.	Collection date is required.	COLL_DATE	lab_col_date_missing
...
2000	Yes	Vital Signs	Vital sign date must be on or before study completion date.	Vital signs date is after the study complete date. Please review entries.	VS_DATE	vitalsigns_date_after_completion

MODIFY THE SAME SPECIFICATION DOCUMENT WITH 'MODIFY'

In order to maintain the attributes in the original ECS, the code in the previous section is modified. Specifically, the section 'write output to Excel' is replaced with the following code.

```

*update the edit check specification based on the results in matchcheck;
data ecs.'editcheck$'n;
  modify ecs.'editcheck$'n matchcheck;
  by id;
run;

```

This above code updates the ECS with the data in data set MATCHCHECK by matching the value of the variable ID (primary key).

Using SAS terminology¹, ECS is the master-data-set and MATCHCHECK is the transaction-data-set. 'ID' is the by-variable and is the common variable in the master-data-set and transaction-data-set. 'MODIFY' uses 'ID' to match observations in MATCHCHECK to ECS.

Furthermore, the libname statements also need modification by adding the option scan_text=NO as follows

```
libname ecs "C:\Users\Nlee01\Desktop\MY SAS\editcheck.xls" scan_text=NO ;
```

This change enables the update of the ECS because the ECS is a Microsoft Excel file². If this option is not added, update to the ECS will fail and a SAS error in the SAS log will show the following error message.

ERROR: No Update/Append allowed. Set libname option SCAN_TEXT=NO to enable Update and Append operation.

Exhibit 4 displays the ECS after it is modified by the code as stated in the 'update the edit check specification' section. The contents are the same as in Exhibit 3 but the original formatting attributes are preserved.

Exhibit 4 (ECS after 'MODIFY')

ID	Programmed	eCRF	Check description	Check Message	Action Field	Check Name
1	Yes	Demographics	Subject age must be at least 18 years old at screening	Subject is younger than 18 years old. Please review entries.	AGE	age_less_than_18
2	No	Demographics	Gender must be provided.	A response is required.	SEX	sex_missing
3	No	Lesions	Lesion number # do not match previous visit	Lesion # at this visit not matching the previous visit. Please review entries.	LESION_NO	lesion_no_not_match_previous_visit
4	Yes	Tumor Assessment	Assessment is marked Yes on the assessment form, Tumor response form must be submitted. If Not Done is not reported then Collection Date is required. This check is optional.	Assessment is answered 'Yes' but there is no matching Tumor Response submitted. Please review entries.	ASSESSMENT	assessment_yes_no_tumor_response_form
5	No	Chemistry	...	Collection date is required.	COLL_DATE	lab_col_date_missing
...
2000	Yes	Vital Signs	Vital sign date must be on or before study completion date.	Vital signs date is after the study complete date. Please review entries.	VS_DATE	vitalsigns_date_after_completion

LIMITATION

There are some limitations with the proposed approach that achieve the outcome in exhibit 4. One of the limitations is the 'bitness' between Microsoft Office and SAS. The 'bitness' must match or the code will not work. For example, if a laptop has SAS 9.2 32-bit for Windows and 32-bit Microsoft Office installed, then the 'bitness' is a match. Although SAS PC files server is a solution to the 'bitness' difference, 'MODIFY' is not supported with PC files server based on my experience.

Another limitation is that one cannot have mixed format in any column in the ECS. For example, the data in the ID column must have the exact same format (i.e., either all are numeric or all are character but not a mix of the two). MODIFY will stop once it encounters the first occurrence of a mixed format (MODIFY still updates the ECS before that point). The same also applies to non-printable characters.

This approach is also limited to one unique record identifier. In other words, it allows one *by-variable* so the master-data-set cannot have more than one unique record identifier.

CONCLUSION

This paper demonstrates a simple way to 'MODIFY' an existing Excel file without the need of creating a new file. The MODIFY statement in SAS serves the purpose. This approach not only streamlines the process of writing SAS output to the same Excel file, it also preserves certain formatting attributes in Excel without additional programming effort. There are some limitations that have been pointed out in this paper. Nevertheless, the use of 'MODIFY' simplifies the body of the code and provides a better option where updates need to be made directly to the original Excel spreadsheet.

REFERENCES

1. SAS Statements: Reference, <http://support.sas.com/documentation/cdl/en/lestmtsref/63323/HTML/default/viewer.htm#n0q9jfr4x5hqsfn17qtma5547lt1.htm>
2. SAS/ACCESS(R) 9.2 Interface to PC Files: Reference, Second Edition <http://support.sas.com/documentation/cdl/en/acpcref/63184/HTML/default/viewer.htm#a002143109.htm>

ACKNOWLEDGMENTS

I would like to thank my colleagues Katrina Paz and Henk Pechler for their support and invaluable input that further enhance the quality of this paper.

CONTACT INFORMATION

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

Name: Nelson Lee, CCDM
Enterprise: Genentech
Address: 1 DNA Way
City, State ZIP: South San Francisco, CA 94080
Work Phone: 6502252121
E-mail: lee.nelson@gene.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.