

Let the Data Step Drive the Report Assembly Line

Sijian Zhang, INTERMACS, University of Alabama at Birmingham

ABSTRACT

Our data quality assurance (QA) reports are sent out to more than one hundred hospitals periodically. All reports have the same components with the information customized to each hospital. The components were assembled for each hospital one by one manually before. Running some simple code in the DATA step, we can speed up the process significantly. It works like generating the reports in an assembly line with limited programmer's intervention.

KEYWORDS

Report, assembly, automation.

BACKGROUND

INTERMACS (Interagency Registry for Mechanically Assisted Circulatory Support) is a national registry for patients who are receiving mechanical circulatory support device therapy to treat advanced heart failure. All the data is collected online through the United Network for Organ Sharing (UNOS) in Richmond, Virginia; and all the analyses and reports are processed in the INTERMACS at the University of Alabama at Birmingham. We receive 36 SAS datasets from UNOS periodically, which include all kinds of information regarding mechanical heart transplantation.

It is common that some of our reports are generated by assembling many components from different sources and using different application programs. If just a couple of ad hoc reports need to be assembled, it is bearable to manually put them together by copy-and-paste. However, if dozens or hundreds of the same kind reports need to be assembled periodically, it will be worth the effort to build up an assembly line to generate those reports in an automatic mass production mode.

INTRODUCTION

The following are four components in the QA report, which is delivered as a password protected zipped folder. Three of them (A, B, and C) contain hospital specific information.

- A. a subfolder of information of clinical summary (ICS) PDF files for all the patients from the hospital;
- B. a subfolder of all the malfunction device report (MDR) PDF files for this hospital;
- C. two PDF files that hold tables and figures of data quality control for this hospital;
- D. two Word files that are the same to all the hospitals.

Figure 1 shows an example report, ABCD.zip opened in WinZip, which is for a hospital coded "ABCD".

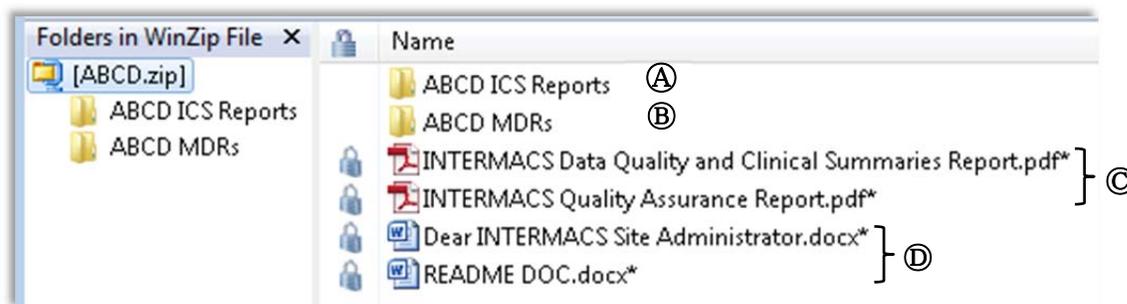


Figure 1

For easy comparison, the following table (Table 1) lists both manual and programming ways that we have experienced for generating the reports.

Table 1: Report Processing Method Comparison

Job	Manual	Programming
Set up folders & subfolders for all hospitals	Set up one by one for the first time, and copy the folder structure later with some adjustment*.	Set up by a SAS [®] program according to current hospital list.
Copy ICS files [Ⓐ]	Go into each source folder, select all the files for one hospital and copy them; then go to the target folder, and paste them.	Done by SAS program according to current hospital list.
Copy MDR folder [Ⓑ]	Go to each source folder, select and copy it; then go to the target hospital folder, and paste it.	Done by SAS program according to current hospital list.
Generate "INTERMACS Data Quality and Clinical Summaries Report.pdf" [Ⓒ]	In Adobe Acrobat Pro, merge two files together by appending.	Merge two PDF files by PDFTK.exe run in SAS according to current hospital list.
Generate "INTERMACS Quality Assurance Report.pdf" [Ⓒ]	In Adobe Acrobat Pro, merge six files together by appending.	Merge six PDF files by PDFTK.exe run in SAS according to current hospital list.
Copy the resulting files in Job 4 and 5	Go to the source folder, select the files and copy them; then go to the target folder, and paste them.	Done by SAS program according to current hospital work list.
Copy two Word files to all the hospitals [Ⓓ]	One-time copy the two files, and paste them to all hospital folders	Done by SAS program according to current hospital list.
Zip each hospital folder with a password	Use WinZip to zip and encrypt each hospital folder	Run WinZip in SAS to zip and encrypt according to current hospital list.

* Some sites may be added or excluded based on the status of the hospitals.

For our last reporting, the QA reports were sent to 109 hospitals. You can imagine what kind of work it could be if all the jobs listed in Table 1 were done by hands. Since there can be dozens of files involved for one hospital, plus the number of repetitions, the attention fatigue can increase the probability of human error. In the programming column of Table 1, "current hospital list" is used for all the SAS programs. It is explained in the next section.

PROGRAMMING

Four kinds of programming jobs need to be done in the assembly line: making folders, copy-and-paste, concatenating PDF files, and zipping folders. Before we get into the coding details, let's look at the "current hospital list".

The current hospital list is a SAS dataset (Site_PWD.sas7bdat), which is generated based on the hospital status information, see an example in Figure 2. This list contains all the hospitals which we will send QA reports to. The variables that are used in the assembly line are Site_Code and PWD.

This dataset offers unique hospital code and password for each QA report. By running Site_PWD dataset through DATA _NULL_ step, the assembly operations will be implemented.



	Site_Code	PWD
1	ABCD	ZnG!SnUz
2	BCDE	Q36EPxv7
3	CDEF	WJd2qi4#
4	DEFG	8YBZ7huV
5	EFGH	U!ZhuYwM

Figure 2

1. Making Folders

Figure 3 displays the result of running the following code.

```
options noxsync noxwait; ❶

%let rptPath=C:\Demo;

%macro createFolder(siteFolder);
x %bquote(md "&rptPath\&siteFolder"); ❷
x %bquote(md "&rptPath\&siteFolder\&siteFolder ICS Reports");
%mend createFolder;

data _null_;
  set Site_PWD;
  call execute ('%createFolder('||Site_Code||')'); ❸
run;
```

- ❶ Option NOXSYNC tells SAS to process as soon as the command is issued. With NOXSYNC in effect, SAS executes an X statement and returns to current session to execute the next statement without waiting. Option NOXWAIT tells SAS that DOS window goes off automatically as soon as the process is finished.
- ❷ X statement is used to execute a DOS command. MD (short for MKDIR) is a DOS internal command that makes a new folder. %bquote() is a macro function to mask the quotation marks.
- ❸ For each iteration in DATA _NULL_ step, CALL EXECUTE routine generates a macro invocation with setting the macro variable &siteFolder to the values of the variable Site_Code in dataset Site_PWD. It means that driven by this DATA _NULL_ step, the macro %createFolder is executed as many times as the number of observations in dataset Site_PWD. The result is that for each hospital which we need to send QA report to, the folder system is made.

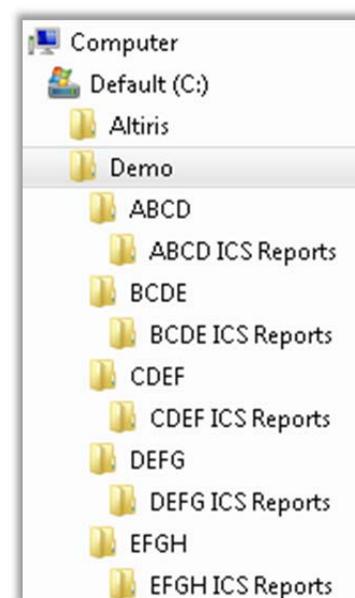


Figure 3

This is the key mechanism whenever we need to process through all the hospitals. Corresponding to the title of this paper, this is how the DATA step becomes the engine of the assembly line.

2. Copy

• For Files:

```
%macro copyFile(file,siteFolder);
x %bquote(copy "&sourcePath\&file" "&rptPath\&siteFolder"); ❶
%mend copyFile;
```

```

%let sourcePath=C:\Master Files;

data _null_;
  set Site_PWD;
  call execute ('%copyFile(Dear INTERMACS Site Administrator.docx' || ',' ||
                Site_Code || ')');
  call execute ('%copyFile(README DOC.docx' || ',' || Site_Code || ')');
run;

```

- ❶ X statement is used here to execute a DOS command COPY. It copies the two Word files (“Dear INTERMACS Site Administrator.docx” and “README DOC.docx”) into all the hospital folders.

- **For Folders:**

```

%macro copyFolder(siteFolder);
x %bquote(xcopy " C:\MDR\output\&siteFolder\&siteFolder MDRs"
              "&rptPath\&siteFolder\&siteFolder MDRs" /i);
%mend copyFolder;

```

❷

```

data _null_;
  set Site_PWD;
  call execute ('%copyFolder(' || Site_Code || ')');
run;

```

- ❷ X statement is used here to execute a DOS command XCOPY. It copies the MDR folder, which contains MDR files, into all the hospital folder as a subfolder.

3. Concatenation of PDF Files

```

%macro mergePDF1(siteFolder);
x %bquote(pdftk "&Cover\Cover1_&siteFolder..pdf"
              "&PDFfiles\&siteFolder..summary.pdf"
              "&PDFfiles\&siteFolder..plots.pdf"
              "&PDFfiles\&siteFolder..plots2.pdf"
              "&PDFfiles\&siteFolder..col.pdf"
              "&PDFfiles\&siteFolder..comp.pdf"
              cat output "&Merged\&siteFolder INTERMACS Quality Assurance Report.pdf");
%mend mergePDF1;

```

❶

```

%macro mergePDF2(siteFolder);
x %bquote(pdftk "&Cover\Cover2_&siteFolder..pdf"
              "&PDFfiles\&siteFolder..pt.pdf" cat output
              "&Merged\&siteFolder INTERMACS Data Quality and Clinical Summaries Report.pdf");
%mend mergePDF2;

```

❶

```

%let PDFfiles=C:\Temp\PDF;
%let Merged=C:\Temp\Merged;

```

```

options xsync noxwait;
data _null_;
  set Site_PWD;
  call execute ('%mergePDF1(' || Site_Code || ')');
  call execute ('%mergePDF2(' || Site_Code || ')');
run;

```

- ❶ PDFTK.EXE is free software by Sid Steward. www.pdfabs.com/docs/install-pdfk is the place where you can download it, and get related information and help. What I did was that I downloaded the zip file for Windows, unzipped it on my computer, and copied the two unzipped files to my system folder. It is a powerful tool to handle PDF files. I used only one of its applications, merging two or more PDF files into a new PDF file.

The command used here is:

```
pdftk file1.pdf file2.pdf ... cat output newfile.pfd
```

- **pdftk** starts the program;
- option **cat**: concatenates the PDF files;
- option **output**: specifies the name of the output PDF file.

In the above DATA step, two new PDF files are generated for all the hospitals. One file is generated by merging six PDF files; another one by merging two PDF files.

4. Zipping Folder with Password

```
%macro WinZip(folder,pwd);
x % bquote (winzip32 -min -a -r -s"&pwd" "&rptPath\&folder..zip"
      "&rptPath\&folder");
%mend WinZip;

options xsync noxwait;
data _null_;
  set Site_PWD;
  call execute ('%WinZip(' || Site_Code || ',' || PWD || ')');
run;
```

- 1 WinZip is popular program for zipping and unzipping files or folders. The one on my computer is WinZip 15.5. Usually people use it in its window style, just like many other applications run on Windows system. However, its command line style applications are also simple and useful, especially for imbedding its applications in other programs.

The command used here is:

```
winzip32 -min -a -r -s"password" zipped-file folder-to-be-zipped
```

- **winzip32** starts WinZip program;
- option **-min**: minimizes the WinZip program window;
- action **-a**: adds contents to the zip file;
- option **-r**: subfolders are included;
- option **-s**: encrypts the zip file with a password.

After this DATA step, every hospital folder is zipped and encrypted with a unique password.

CONCLUSION

Just as the SAS macro is considered when there are many SAS code repetitions, the batch processing approach is worth trying when we need to handle the same processing procedures repeatedly. Sometimes, the operation functions are not internal to SAS. However, since for most of programs, including SAS, command line execution syntaxes are available, by taking the advantage of SAS DATA step and X statement, we can easily automate those processes. Once the coding for automation is done, the benefit is not only the current job processing efficiency, but also the scalability of the operation. If one day in the future we need to assemble five hundred QA reports, I will do the same thing as what I do for assembling one hundred reports: clicking on Run button.

REFERENCES

1. Sid Steward, www.pdf labs.com/docs/install-pdf tk
2. David Brennan, A Quick Guide to the WinZip Command Line Add-On, Paper CC01, PhUSE 2006, <http://www.lexjansen.com/phuse/2006/cc/cc01.pdf>

CONTACT INFORMATION

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

Sijian Zhang, MD, MS, MBA
Database Administrator

Data Coordinating Center
INTERMACS

Division of Cardiothoracic Research
Department of Surgery
University of Alabama at Birmingham
sijian@uab.edu



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.