

Template based file generation

Christoph Baumer, Syneos Health, Berlin, Germany

ABSTRACT

In this poster, the use of templates for the data driven generation of Word™ documents, programs and statistical outputs is presented. The latter uses an ASCII shell-file containing a dummy layout of the output. This layout file contains sections for titles and column headers, output content and footnotes. In the resulting ASCII output file, the different parts are automatically replaced with the required data. Creation of word documents works in a similar fashion, although the functionality needs to be broader. This ranges from simple replacement of tags to full insertion of sections from other documents and flexible generation of tables from external data sources. The creation of programs follows a simpler approach, since typical replacements are often restricted to libraries, datasets and variable names, but also insertion of code parts based on meta-data conditions can be realized. These different approaches are presented to display similarities and differences, together with various tricks and tips.

INTRODUCTION

The examples presented in the poster display different applications of using templates as the basis for file creations. The technique as such is a text replacement of varying complexity, using different ways of representing the “to-be-replaced” parts. The first example almost fully consists of parts that are going to be replaced, the word document example contains a rather small amount of text to be replaced (compared to the full document) and the last case of the SAS program templates shows only the replacement of key parts of the code.

The poster has its focus on displaying the concept and is not intended to provide implementations of the actual solutions. Even though similar in the underlying idea, the different examples require very different implementations which can become very complex. The functionalities presented here are by no means a complete representation of what can be done and you might want to adapt your solution based on your own requirements. However, what I see as one major advantage of the concept is the high level of expandability. Once you have the main program running, it is rather easy to add further features, since they are based on the same concept.

STATISTICAL OUTPUTS

Statistical outputs can be created in different ways. Many pharma companies have their own reporting system, typically consisting of a various number of macros or the outputs can be created using SAS™ procedures such as PROC REPORT, PROC TABULATE or, not the best idea, PROC PRINT. With the presented approach (actually just being a specific kind of reporting system), you have a very high level of control over the layout of the output. Further, the approach is very visual, and is thus much more intuitive than using, say, PROC REPORT. Using macros for creating the headers and footers, as the poster illustrates, means you have central control over the outputs, making it easier to make changes for all outputs at once, which is a big advantage over having many single programs using PROC REPORT. The obvious disadvantage (of any reporting system as such) for the company is that each programmer joining needs to be trained on the system. On the other hand, the disadvantage for the programmer is that when leaving the company the skills acquired for the system can only minimally be transferred on to the next position.

The technical background of the approach is to use a DATA _NULL_ step together with FILE and PUT statements to write the content of the report dataset to a flat text file. Of course the details can be tricky, since the correct line numbers, page breaks and indentations need to be calculated properly.

The outcome of this approach is an ASCII file, which means there is further processing required to create RTF type files which can be used for the Clinical Study Report. This task can be done using Visual Basic Macros, but since, strictly speaking, RTF files are also plain ASCII files, SAS too can easily be used to add the RTF markup parts to the resulting files making the outputs proper RTF files. To have the outputs as simple text files can also be very helpful actually, if one wants to compare for example the current with the previous results, using a simple text comparison tool like for example *WinMerge™* or *DiffMerge™*.

The presented approach is more recommended for CROs than for pharma companies, since CROs need to be more flexible when it comes to the layout of the outputs, because different clients might have very different requirements on how the deliverables should look.

PhUSE EU Connect 2018

MS WORD DOCUMENTS

The approach for Word documents is different in the sense that not only one “model” page is provided as the template, but a full Word document needs to be prepared beforehand. The document contains tags in a certain format, which will be changed by performing various kinds of text replacements. The implementation uses a combination of SAS and Java™, specifically the Aspose.Words™ Java library. Since MS Word documents are not so easy to handle with other programs, there are not many alternatives available, especially not when working on a UNIX™ environment. Often there are just no alternatives left but to produce such documents by hand.

A typical use case for this automatic file generation of Word documents is validation documentation. In the poster example, parts of a test protocol used for macro validation is shown. For this kind of task, it needs to be considered whether this is a repetitive task which occurs often enough to justify a programmatic solution. Completing a Word document by hand can often be done fast enough, depending on the number of entries required.

If, due to SOP updates, the underlying document template changes frequently, it might also be a better idea to always go the whole way “by hand”, to avoid the inadvertent usage of outdated documents. On the other hand, if you are in the process of validating a high number of macros, it makes a lot of sense to automate the documentation together with the validation. And of course the application of using word templates is not restricted to validation documentation only. You can use the technique to automatically create any kind of MS Word document, from program specifications to user manuals. It is especially helpful when you have text passages that are intended for different documents, to maintain consistency over the whole documentation of your project.

SAS PROGRAMS

As shown in the examples, the amount of SAS code required can range from just one line stored in a data step variable up to full programs stored in separate files. The alternative for the simple cases would be to use concatenated expressions like:

```
cmd = "select ' "||strip(ds)||
      "', usubjid, count(*) from resultds."||strip(ds)||
      " group by usubjid;"
```

The main advantages of the approach displayed in the poster over using such concatenations are readability and maintainability. Just two variables used in such a concatenated text string / variable combination can make the code look messy and unreadable. The presented approach is especially recommended when it is expected that the code will be extended and/or frequently changed in the future.

CONCLUSION

Even if the presented examples are very different in their applications, one common point to take away is the very high level of expandability. It can be easy to start with a rather small set of basic functionalities and you can keep expanding as you identify new requirements and possibilities. Of course it is always crucial to consider alternatives, as the presented approach does require a certain amount of effort, at least for the initial set up. But once set-up, it is easy to go further and further with the approach and it can enable you to automate big parts of your programming and documentation.

ACKNOWLEDGMENTS

Many thanks go to Yuliia Bahatska, Katja Glass, Rowland Hale and Kathryn Wright for their ideas, reviews and recommendations.

CONTACT INFORMATION

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

Christoph Baumer
Syneos Health
Joachimsthaler Strasse 10-12
10719 Berlin, Germany
Email: christoph.baumer@syneoshealth.com
Web: www.syneoshealth.com

Brand and product names are trademarks of their respective companies.