

UCF Stored Process Conversion for Current STEM Retention Reports

Carlos Piemonti, UCF

ABSTRACT

At the University of Central Florida (UCF), in order to reduce the proliferation of similar reports in our SAS® Information Delivery Portal, we have been tasked with generating different multiple reports from one stored process. Users will be prompted to select multiple criteria which will determine the report to be created, as opposed to having multiple stored processes and running them separately.

INTRODUCTION

UCF's reporting needs are as varied as any other academic organization. At UCF, Institutional Knowledge Management (IKM) is tasked with providing actionable information regarding our University to internal and external customers alike. The primary resource to facilitate all of UCF's reporting needs is our SAS® Business Intelligence environment, and most recently our SAS® Visual Analytics suite.

To help in this endeavor IKM provides the resources when it comes to report generation and support required to disseminate this information to our consumers.

During the last seven years IKM developed over 250 new SAS® Stored Processes to better serve a very specific internal demand of data from any full-time faculty, staff or executive users across campus. The purpose of any program created is to answer one or multiple questions at a time, but some of those packages invite users to formulate more questions, and the code tends to evolve inevitably once consumers are exposed to the data generated.

Recently users from one of the biggest units on campus requested new reports, in this case, very similar to a couple of Stored Processes already available in our SAS Information Delivery Portal. In this sense, this paper will show in detail how we handled this case, challenges faced, design, implementation, and results.

CASE OF STUDY

Retention Reports are essential. IKM provides a broad variety of these reports. This big unit on campus was already receiving among others, the following ones: 'STEM Undergraduate Retention Report at UCF' and 'STEM First Generation and Low Income Undergraduate Retention'. They recently requested two new reports, 'STEM First Generation Undergraduate Retention' and 'STEM Low Income Undergraduate Retention', to be incorporated in their access. Please note STEM is an acronym that refers to the academic disciplines of Science, Technology, Engineering and Mathematics.

Initially it was considered to create two new Stored Processes based on the original ones and add those to the Portal. But adding more Stored Processes to our already convoluted Portal could cause more scrolling and confusion rather than to help. As a consequence it was considered to consolidate these similar scenarios in a single package.

CHALLENGES

It is not easy to use somebody else's code. Especially if it was written few years back. Syntax evolves, even our own way to write code gets tweaked over time. If our own code is not commented to help in understanding what it is intended to do, it could be really difficult to follow few years later even for ourselves. A step above in difficulty could be inheriting code from somebody else. As a good practice, in-house developers always instructed me to make good notes in my code and in any of my documentation. Gladly, the original code had really good notes from the original programmer and it was very straight forward to follow.

Then questions arise like 'How to prompt users?' The idea was having them select each report individually after running the Stored Process. But, 'would all these reports work if they don't share the exact same prompts for all the specific options the original reports already had?' Prompts like 'Student Type', 'College', 'Cohort Type', 'Gender', 'Ethnic Group' and 'Student Group' were required.

Initially we thought of creating a lookup table in case these prompts were not the same, but after realizing these prompts were actually the same for all the reports then there was no need for making things look complicated.

DESIGN & IMPLEMENTATION

Having all those similarities, lead us to the idea of combining all the code in one package and adding a new prompt for selecting the reports in the actual Stored Process, being added as an additional one being called before the ones already in place.

The next step was to create the new code for the new reports. The original code was copied and modified in order to get these new reports working. A couple of separate SAS programs were created using SAS® Enterprise Guide® 7.1. Few joins were tweaked, restrictions were restructured, titles and footnotes included in the code were also updated to reflect the correct output, in this case, PDF files.

Each program (original ones and new ones) had the following basic structure:

```
%global;
options;
libname;
%include;

%macro retent;
BLOCK OF CODE;
%end;
%mend;
%retent
```

Now we had to put together all four blocks of code and make a call for each report using 'if' statements, being careful enough to insert all 'if' statements inside the macro, leaving only common code elements outside. The new structure was created as follows:

```
%global;
options;
libname;
%include;

%macro retent;

%if "&ReportName" = "STEM Undergraduate Retention at UCF" %then
%do%;
BLOCK OF CODE 1;
%end;

%if "&ReportName" = "STEM First Generation Undergraduate Retention" %then
%do%;
BLOCK OF CODE 2;
%end;

%if "&ReportName" = "STEM Low Income Undergraduate Retention" %then
%do;
BLOCK OF CODE 3;
%end;

%if "&ReportName" = "STEM First Generation Low Income Undergraduate Retention"
%then
%do;
BLOCK OF CODE 4;
%end;

%mend;
%retent
```

RESULTS

After putting together all the pieces and running the new Stored Process, a comparison was made with the new output versus the former output once we selected the same parameters. The results were as expected: same reports. Also titles and footnotes were verified. For the new reports, we had no data to compare with other than the data being

generated in EG that confirmed our results were the same, and after deep analysis they were also in between the expected ranges of the data collected. Also titles and footnotes were vetted.

When running the new Stored Process prompts are being displayed as shown in Display 1, below.

STEM Undergraduate Retention Reports at UCF

Bookmark Publish E-mail

☐ Show only required items (denoted by *)

Parameters

*Please select a STEM Report from the list:

*Student Type:
Parameter for student type selection.
FTIC

*College:
Parameter for college selection.
All Colleges

*Cohort Type:
Parameter for cohort type selection.
Summer-Fall Full-Time

*Gender:
Parameter for gender selection.
Male and Female

*Ethnic Group:
Parameter for ethnic group selection.
All Ethnic Groups

*Student Group (Example: RHNR):
Parameter for entry of student group
All Student Groups
PeopleSoft Student Group Code

Run

Display 1. STEM Undergraduate Retention Reports at UCF

Find below (Display 2) a detail of the prompt to select a report.

*Please select a STEM Report from the list:

STEM Undergraduate Retention at UCF

*Student STEM First Generation Undergraduate Retention

STEM Low Income Undergraduate Retention

STEM First Generation Low Income Undergraduate Retention

Display 2. Prompt to select STEM Reports

FUTURE DEVELOPMENT

Possibly, the next step of improvement of these reports is an overhaul of the entire PDF layout that is being developed by a colleague. If the concept gets approved, we might have to apply this new layout to all our in-house retention reports.

CONCLUSION

It is natural over time for objectives and questions to be subject to change, to disappear, or to be tweaked. New developing tools also invite users to be more daring in the kind of data they would like to be receiving, and at the same time, users inspire the creation of new tools and new better ways to serve others thanks to the 'what if' that always arises.

As developers we are usually exposed to different challenges that probably go hand-in-hand with the constant evolution of technical tools. Also we are affected in how data is being handled by users and other developers in the general community at any time. We are all always subject to change, adapt and evolve, or simply disappear.

As shown in this paper, it is possible to reduce the proliferation of similar reports in our SAS Information Delivery Portal by combining multiple (but similar) programs in one stored process simplifying access for the users.

REFERENCES

Cody, Ron. 2007. *Learning SAS® By Example: A Programmer's Guide*. 626 pages. Cary, NC: SAS Institute Inc.
<http://support.sas.com/>

ACKNOWLEDGMENTS

Sincere gratitude and appreciation is extended to Ms. Evangeline Collado and Mr. Scott Milbuta for all their input and support during my career as a programmer in SAS BI and SAS VA environments.

CONTACT INFORMATION

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

Carlos Piemonti, MBA
Database Analyst/Programmer
University of Central Florida
Institutional Knowledge Management
12424 Research Pkwy, Suite 215
Orlando, FL 32826
E-mail: Carlos.Piemonti@ucf.edu
Web: <https://ikm.ucf.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.