

SESUG Paper 268-2018
**A Beginner's Take On Conditional Logic Using SAS
Enterprise Guide® 7.15 and SAS® Macros.**

Mykhael Walker, University of Central Florida

ABSTRACT

Learning any new system can be a daunting task, but learning a system that has multiple ways to solve the same problem can be refreshing and frustrating at the same time. When first introduced to SAS Enterprise Guide® 7.15, all the available features can be overwhelming. Once the initial shock subsided, it was time to put the tools to work. A task that commonly comes up when trying to solve a problem is the following scenario: If A happens, then do B...conditional logic. SAS Enterprise Guide 7.15 provides a way for the user to manipulate conditional logic visually and SAS® code provides multiple ways to manipulate conditional logic via code. This paper will discuss how to use the conditional logic capability built in SAS Enterprise Guide 7.15 to solve an issue along with alternatively using macros in SAS code to solve the same issue, but with a little more flexibility.

INTRODUCTION

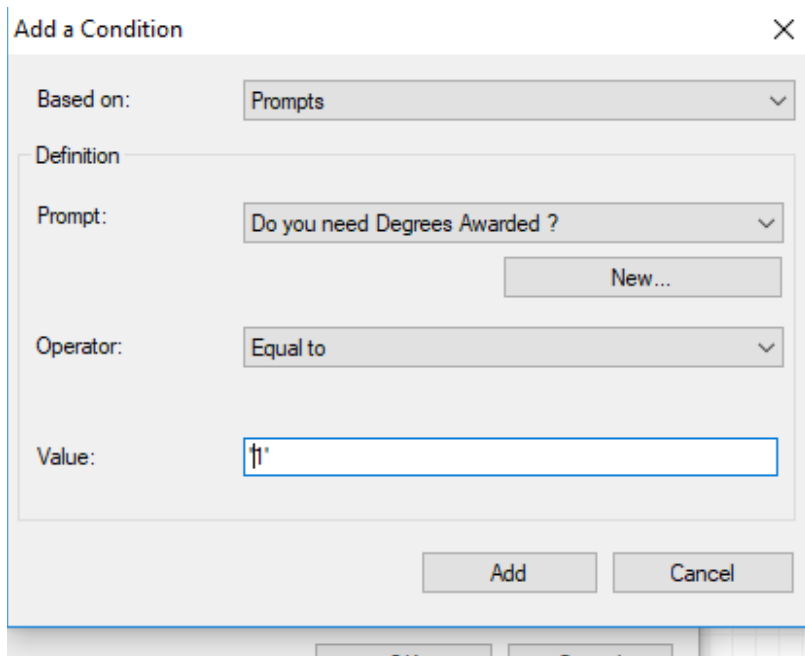
In SAS Enterprise Guide®, there are several different ways that conditional logic can be applied in a project via case statements in an advanced expression, recoding of variables, as well as to control task execution. To some who use SAS Enterprise Guide on a daily basis the ends and out of how to manipulate data with the tools that are provided in SAS Enterprise Guide are apparent. One of my first tasks as a developer within the Office of Institutional Knowledge Management was to help a user who was having difficulties navigating and executing Enterprise Guide projects. These Enterprise Guide projects were created so that the user would essentially be able to easily provide the data sets necessary for routine information data requests that came into the office. However, the user found it difficult to run the projects necessary to obtain the data to satisfy the data requests. After meeting with the user, I was able to determine that the user was having difficulties reaching the “final data set” needed because they were unable to follow the multiple branches of a project. This paper will discuss how SAS Enterprise Guide is used to simplify a complicated project into one that utilizes project conditions to control task execution based upon prompt values allowing anyone regardless of knowledge of the underlying logic to run the project and retrieve the desired results. The discussion will also include how to take the same conditional logic used in the previous scenario and implement them in a SAS® program that conditionally executes code utilizing SAS® macros.

CONDITIONAL LOGIC USING SAS ENTERPRISE GUIDE

In SAS Enterprise Guide, as a project is being constructed, multiple runnable objects are created until a “final data set” is reached. Often times there can be different variations of these “final results” in a project dependent upon the specifications a user is looking to obtain. Instead of creating several different projects for each specification, we can create one project that will produce a result based upon certain specifications. To do this we can utilize the conditional logic feature in enterprise guide to add conditional logic to a runnable object within a process flow so that we only obtain the “final data set” we desire. SAS Enterprise guide makes it simple to achieve this level of “if-then-else” conditional logic for controlled task execution by allowing a user to right click on a runnable object and select add a condition. In order to solve the task of simplifying a complicated project I chose to utilize both processing conditions and prompt values. A processing condition can be based on a variety of different conditions but in order to solve my issue the processing conditions I would use needed to be based on prompt values. Therefore, the first true step in solving the task was to analyze the project and determine what different questions needed to be formulated in order to create meaningful prompts. The prompts play a major key, as they will ensure that the necessary runnable objects execute and the desired result is produced. In SAS Enterprise Guide, the prompt manager creates the prompts that will be used throughout a project. In

order to create processing conditions, SAS Enterprise Guide provides a useful conditional processing wizard that walks the user through the creation of project conditions. Using the prompts created via the prompt manager, we are able to complete the conditional processing wizard.

The first step in the conditional processing wizard is to add a condition that evaluates to true as seen in Figure 1.



The screenshot shows the 'Add a Condition' dialog box. The 'Based on:' dropdown is set to 'Prompts'. The 'Definition' section contains a 'Prompt:' dropdown set to 'Do you need Degrees Awarded ?', a 'New...' button, an 'Operator:' dropdown set to 'Equal to', and a 'Value:' text box containing '1'. The 'Add' and 'Cancel' buttons are at the bottom.

Figure 1. Condition Based on Prompts.

Next, you want to designate the task you would like to have processed if the condition is true as well as designate the task you would like to have processed if the condition is not true as seen in Figure 2.

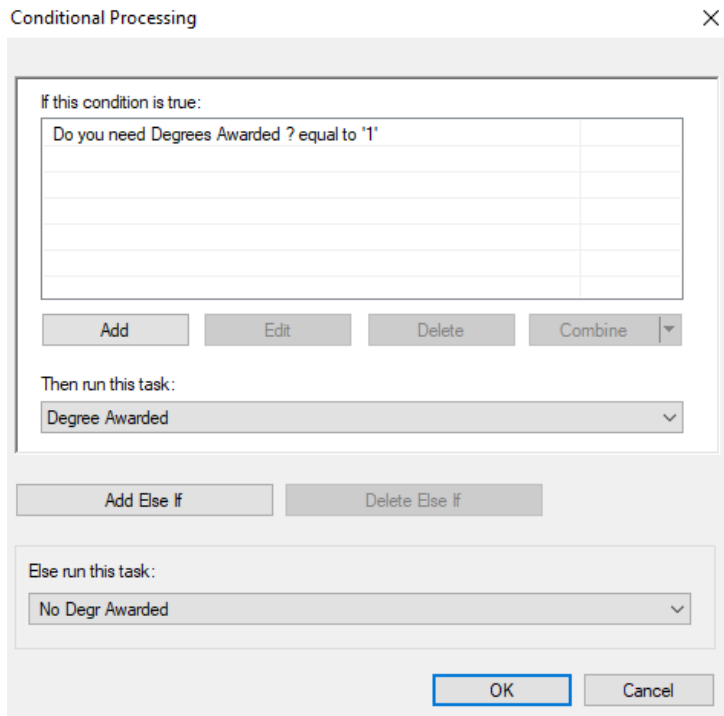


Figure 2. Conditional Task Execution.

Once you have confirmed the conditions and designated tasks you would like to have executed, SAS Enterprise Guide places condition indicators on the tasks affected by the processing condition. This gives the user a visual representation of the processing logic so the user can see exactly what step in the project will be affected by a processing condition. In Figure 3, you can see an example of the visual display SAS Enterprise Guide applies to objects with processing conditions.

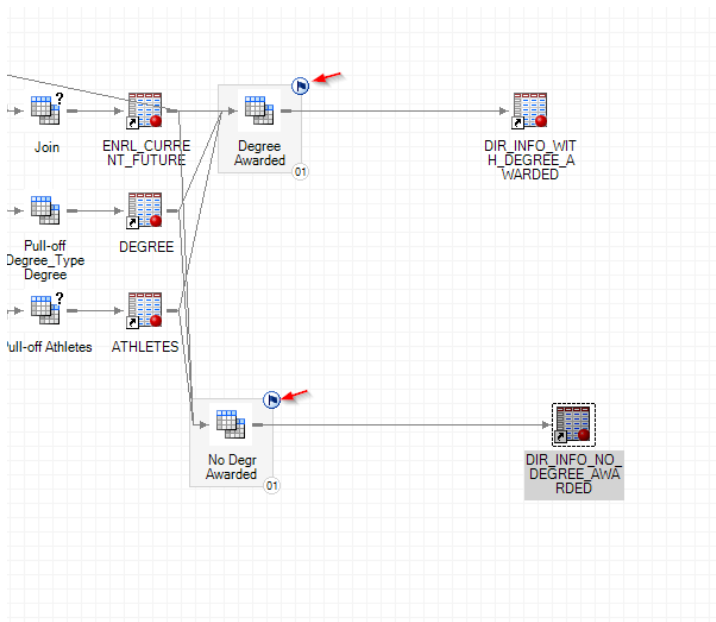


Figure 3. Processing Condition Visual Display

With processing conditions, you can gain control on how a project is executed, and the nice thing about SAS Enterprise Guide is that it makes it easy for the end user to create these conditions as well as visually see how the data is constructed via the process flow. As it pertains to my initial task of simplifying a complicated SAS Enterprise Guide project, the benefit of adding the processing conditions to the complicated projects could be seen almost immediately. The user who at first struggled to determine what “final data set” was needed was now able to run a once complicated project as if they were the creator of the SAS Enterprise Guide project.

CONDITIONAL LOGIC USING SAS MACROS

As with many things in SAS, there are multiple ways to solve the same problem; it all depends on the developer and the tools that they are adept at using. A couple of the tools most SAS programmers are well aware of are SAS Macros and SAS Macro variables. A macro variable can be viewed as a data variable that is always a character and does not belong to a data set. A macro, on the other hand, is a compiled program that is called when a SAS program is executed. For the scope of this paper, I will not be delving into the details of how each can be used or structured. However, I will be providing an example of how I used both SAS Macros and SAS macro variable to solve the same problem as discussed earlier in this paper.

CREATING MACROS

For the macros you intend to create to have any meaning, you need to create the tables that would eventually lead to your “final data set.” These tables could be looked at as the multiple branches that can be seen throughout an Enterprise Guide project. The macros that you create would each represent a different final data set; these macros would then be called later on in the code. The code below is an example of a macro definition used:

```
%macro stdnt_w_degr_no_maj;

PROC SQL;
    CREATE TABLE WORK.STDNTS_WITH_UCF_DEGREE_NO_MAJ AS
    SELECT DISTINCT
        ...
    FROM WORK.DEL_TEMP_STDNTS t1
        LEFT JOIN WORK.DEL_DEGREES t2 ON (t1.EMPLID = t2.EMPLID) ORDER
    BY t1.PERS_PRIMARY_LAST_NAME, t1.PERS_PRIMARY_FIRST_NAME;

%mend stdnt_w_degr_no_maj;
```

Multiple macros would be defined like the above, based upon the different outcomes that are needed.

IMPLEMENTING PROCESSING CONDITION

Based on the desired “final data set” there needed to be some sort of conditional logic implemented that would rely on the input of the user and execute the necessary macro based upon the user's input. To mimic the if then else logic used via the SAS Enterprise Guide processing condition wizard, you can use a combination of a _NULL_ DATA step and a CALL EXECUTE routine like in the following example:

```
DATA _NULL_;

    if &need_degree = '1' and &need_major = '1' then call execute
        ('%stdnt_w_degr');

    if &need_degree = '1' and &need_major = '0' then call execute
        ('%stdnt_w_degr_no_maj');

    if &need_degree = '0' and &need_major = '1' then call execute
        ('%stdnt_no_degr');
```

```
if &need_degree = '0' and &need_major = '0' then call execute  
('%stdnt_no_major');  
  
run;
```

The macro variables referenced in the above code represent the values returned to the code once a user has answered the prompts provided. From the above, you can see that it is possible to create many conditions that can then be used to determine whether a macro should be executed.

CONCLUSION

As with most problems that are solved programmatically, there are usually several paths that can be taken that all lead to the same destination. I often find in SAS that the aforementioned statement holds true, one developer may use a certain tool or method to solve a problem while another developer may solve the same problem with a very different tool or method. As showcased in this paper, SAS Enterprise Guide provides an intuitive way to handle conditional logic and users can use those tools to control program execution. The same methodologies utilized in SAS Enterprise Guide to control program execution can be implemented with SAS macros and SAS macro variables providing the developer with another alternative (tool) to accomplish the same goal.

REFERENCES

Slaughter, Susan J. and Lora D. Delwiche. 2010. *The Little SAS Book for Enterprise Guide 4.2*. Cary, NC: SAS Institute.

Slaughter, Susan J. and Lora D. Delwiche. 2004. "SAS Macro Programming for Beginners." *SUGI 29 Proceedings - Tutorials*, Montreal, Canada : . Available at <http://www2.sas.com/proceedings/sugi29/243-29.pdf>.

CONTACT INFORMATION

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

Mykhael Walker
Decision Support Developer
E-mail: Mykhael.Walker@ucf.edu
University of Central Florida
Institutional Knowledge Management
12424 Research Pkwy, Suite 215
Orlando, FL 32826
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.