

How to create a working outline for SAS Global Forum

A working outline is an outline you use for developing your paper. This is the outline where you lay out the basic structure of your paper. You must have a general and specific purpose; an introduction, including a grabber; and a concrete, specific purpose. You also need about three main points and a conclusion.

One strategy for beginning your working outline is to begin by typing in your labels for each of the elements. Later you can fill in the content.

When you look ahead to the full-sentence outline, you will notice that each of the three or so main points moves from the general to the particular. Specifically, each main point is a claim, followed by particular information that supports that claim so that the audience will perceive its validity. For example, for a paper about joining observations in Base SAS using Proc SQL, your first main point might focus on the idea that Proc SQL can be more efficient than the Merge statement. You might begin by making a very general claim, such as “One major advantage of using Proc SQL comes from the fact that datasets do not need to be explicitly presorted as Proc SQL will implicitly do the sorting for you,” and then become more specific by providing examples, syntax, or examples to support your primary claim.

A working outline allows you to work out the kinks in your paper, but more importantly it allows for paper reviewers to see the potential of the finished product. The working outline shouldn't be thought of a “rough copy,” but as a careful step in the development of your message. It will take time to develop. The working outline will also help ensure that your abstract and your paper tie into one another.

Adapted from <https://2012books.lardbucket.org/books/public-speaking-practice-and-ethics/s15-02-types-of-outlines.html>

Example of creating a working outline

To help illustrate how to create and use a working outline, we've created the following multi step examples below from an actual paper. [Step 1](#) is the only required outline to submit for review.

However, the outline created in [Step 2](#), while not required, provides the reviewer a better guide to what the paper will be about. [Step 3](#) is also optional but will help the paper take form. It may also help to identify other images or examples which will help illustrate the topic. As the abstract is usually completed at this point, it's good to include it in this version of the outline as a guide to the paper. Once Step 3 is completed, [Step 4](#) is to revise and complete the paper by filling in the additional information.

Step 1 – General outline to follow (Required)

- I. Abstract
 - II. Introduction
 - a. Include the grabber to hook in the reader
 - III. Topic 1
 - a. General point
 - b. Detailed explanation
 - IV. Topic 2
 - a. General point
 - b. Detailed explanation
 - V. Topic 3
 - a. General point
 - b. Detailed explanation
 - VI. Conclusion
 - VII. References
 - VIII. Contact Information
-

Step 2 – Add in more information (Optional)

- I. Abstract (*Typically already completed prior to paper creation*)
 - a. Topic: techniques for vertically combining datasets
- II. Introduction
 - a. Grabber: A SAS programmer with tasks requiring to vertically combine data
 - i. Why?
- III. Process of Appending
 - a. A generalized explanation of appending.
- IV. Traps of appending
 - a. Appending data can cause problems
 - b. Problems:
 - i. Discrepancy in datatypes
 - ii. Discrepancy in variable lengths
 - iii. Discrepancy in variable formats
 - iv. Variables aren't the same between all datasets
 - v.
- V. Best Practices
 - a. Before appending
 - b. Resolving

- c. Validation
 - VI. Methods
 - a. Data set statement
 - b. Obstacles
 - i. Efficiency
 - c. Proc Append
 - i. Force
 - 1. Efficiency
 - d. Proc Sql
 - e. Proc Datasets
 - VII. Method Comparison
 - a. Is one better than the other?
 - VIII. Conclusion
 - a. Quick summary of all points
 - IX. References
 - X. Contact Information
-

Step 3 – Move from outline to paper with beginning of sentences, bullet points and placeholders (Optional)

Abstract

Although not as frequent as merging, a data manipulation task which SAS programmers are required to perform is vertically combining SAS data sets. The SAS system provides multiple techniques for appending SAS data sets, which is otherwise known as concatenating, or stacking. There are pitfalls and adverse data quality consequences for using traditional approaches to appending data sets. There are also efficiency implications with using different methods to append SAS data files. In this paper, with practical examples, I examine the technical procedures that are necessary to follow to prepare data to be appended. I also compare different methods that are available in BASE SAS to append SAS data sets, based on efficiency criteria.

Introduction

A SAS programmer may encounter different task situations on a given project where the need to append data sets arises.

- Time period series analysis might be a cause to need to do this append.
- Specifically this paper: In these examples, the purpose of appending is due to the need to analyze data over a longer time period, such as a full year, or to look at trends across a 3 or 5 year time span.

Process of Appending

Appending, concatenating, and stacking are all terms used which refer to vertically combining SAS data sets.

[flowchart]

The Traps of Data Set Appending

- Programmer responsible for data integrity

Data sets can have discrepancies in variable types, variable lengths, and variable formats. Another trap to lookout for is when the data sets don't share the same set of variables.

Discrepancy in Variable Types

- Short explanation

[show screenshot of log]

Discrepancy in Variable Lengths

- Short explanation

[show screenshot of log]

Discrepancy in Variable Format

- Short explanation

[show screenshot of log]

Variable Mismatches

- Variables can be in one data set but not the other.

[show screenshot of log]

Best Practices

Before Appending

In order to ensure consistency between the data sets you're appending its necessary to examine the contents of those data sets, the variables they contain, and the attributes of those variables.

- Proc Contents
 - Show Code
 - Show output example
- Proc Datasets
 - Show code
 - Same results with both
- Proc Compare
 - Show Code

Resolving the Inconsistencies

- Different functions can be used to standardize the data such as input, put, length, etc.

Code Example:

```
Data Sales_95;  
  Length State $22;  
  Set appsds.Sales_95 (Rename=(Mon=MonYR Quarter=Qtrc State=NewState Actual=Actualn)  
  Drop=County);
```

```

*Re-assign State with new length;
State = NewState;
*Convert Actual to numeric;
Actual = Input(Actualn, dollar12.2);
*Convert MonYR to character;
Quarter = Put(Qtrc, 3.);
Drop NewState Actualn Qtrc;

```

Run;

After Appending

Validate the data by using Proc Contents or Proc Freq (for example).
[proc freq code]

Methods of Appending

Data Step Set Statement

I consider the DATA STEP SET statement to be the traditional approach to appending data sets.
[show code]

Obstacles to Appending

- List things that might cause SAS to error

Discrepancy	Result	Note
Variable Type	Will not Append	Error Message
Variable Length	Will Append	Variable length in first data set carried over. Issues Warning Message
Variable Format	Will Append	Variable format from first data set carried over.
Variables	Will Append	Missing values for records from data set without variable

Efficiency Implications

The DATA STEP can be a time-consuming and resource intensive construct.
[show log with # of observations and time spent]

Proc Append

- Other option is proc append
- Limited to two data sets
- Straightforward code
- [show code and log entry]
- Won't append if discrepancies between data sets

The Force Option

Using the FORCE option, PROC APPEND will combine data sets if discrepancies exist in the variable type, length, or variables between the BASE and DATA data sets

SAS® GLOBAL FORUM 2020

Trap	Result	Note
Variable Type	Will append with FORCE option. Issues warning message.	Missing values for records from DATA data set.
Variable Length	Will append with FORCE option. Issues warning message.	Length from BASE data set prevails. Records from DATA data set may contain truncated values.
Variable Format	Will append w/warning msg.	Format from BASE data set prevails.
Variables	Will Append with FORCE option. Issues warning message.	Missing values for records from data set without variable

[show code and log entry]

Efficiency Implications

- Improved efficiency

[show code and log entry]

Proc SQL Set Operators

- You can create a vertical append in proc sql using set operators
- There are four different SET operators in PROC SQL; INTERSECT, EXCEPT, UNION, and OUTER UNION
- User Outer Union

[show code and log entry]

Proc Datasets

- Similar to proc append

[show code and log entry]

Method Comparison

- Compare all the processing time differences between all the examples and give an explanation of opinion

BASE SAS Method	Real Time	Ranking	% Change	CPU Time	Ranking	% Change
DATA STEP SET statement	50.36	4		5.34	3	
PROC APPEND	27.33	2	-46%	2.80	2	-48%
PROC SQL OUTER UNION	48.91	3	-3%	8.00	4	50%
PROC DATASETS	22.85	1	-55%	2.74	1	-49%

SAS® GLOBAL FORUM 2020

Conclusion

- Always validate your data outcomes
- In my test cases, PROC DATASETS slightly outperformed PROC APPEND in processing time.

References

Contact

Step 4 – Fill in the blanks and complete the paper (Required)