MWSUG 2016 — Paper TT14

Ron Fehd, SAS-L’s Macro Maven, Answers Your Macro Questions

Ronald J. Fehd, Stakana Analytics

Abstract

Description: SAS® software consists of two languages, SAS and its macro language.

Purpose: The purpose of this talk is to provide overview and perspective of how SAS works and how the macro language can work both within and before SAS program statements and steps.

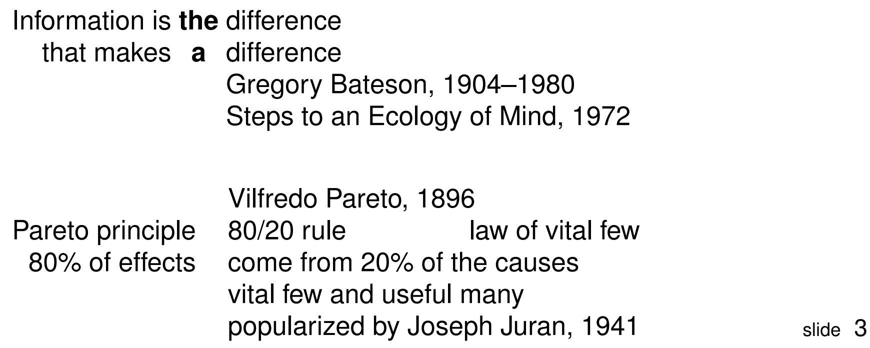
Audience: all levels

Keywords: macros, autoexec, conditionals, configuration, compile, execution, loops

In this paper

Theory

What is Information?



Information is **the** difference
that makes **a** difference
Gregory Bateson, 1904–1980
Steps to an Ecology of Mind, 1972

Vilfredo Pareto, 1896
Pareto principle 80/20 rule law of vital few
80% of effects come from 20% of the causes
vital few and useful many
popularized by Joseph Juran, 1941

slide 3

5 aspects of learning new software

1. variables	constants	global symbol table
2. control	boolean comparisons, logic	stop additional code branching
3. loops	boolean: exit on condition	while until
	list of items	
	iterative	start, stop, step sequence: step eq 1 series: step ne 1
4. functions		process or procedure
5. syntax		

slide 4

HIPO, three aspects of every program

Hierarchical	location	type	calls
	top	module	routines
	middle	routine	subroutines
	bottom	subroutine	
Input			
Process		process	procedure
		N steps	Q: if . . . else
	executed	always	conditionally
Output			

slide 5

data structure or algorithm?

20%	80%
compiled	executed
data structure	algorithm
object	method
`attribute`	

$(20\%)^3 \cong 0.8\%$

slide 6

How SAS works

startup process

1. sas.exe		
2. sasv9.cfg	allocate environment variables assign options	
3. command-line	options	*.cfg sysparm
4. autoexec	add to / change, entries in global symbol table	
5. initstmt		initial statements
6. program		
7. termstmt		termination statements

slide 7

autoexec process

adding entries to global symbol table Fehd [7]

environment variable(s)	site_root	
filenames	project, site_includes, site_macros	
libname	library	
options	sasautos=(project site_macros sasautos)	
running text:	titles	footnotes
macro	variables	definitions

slide 8

Conditionals

Writing Testing-Aware Programs Fehd [1]

if fail then stop

```
%let data = sashelp.class;
%sysfunc(ifc(%sysfunc(exist(&data))
            ,%nrstr(%put info:     exist &=data;)
            ,%nrstr(%put fail: not exist &=data;
                    endsas;) ))
```

slide 9

additional statements

```
DATA process;
     attrib ...;   * <---<<< define data structure;
*...;
run;
*** echo data structure to log?;
%sysfunc(ifc(%sysfunc(getoption(source2)) eq SOURCE2
            ,%nrstr(proc sql;
                    describe table &syslast;
                    quit;) , ))
```

slide 10

conditionals in macro

```
%macro demo(data    = sashelp.class
           ,testing = 0 );
%let testing = %eval(   not(0 eq &testing)
                     or %sysfunc(getoption(mprint)) eq MPRINT);
*...;
%if &testing %then %do;
    proc sql; describe table &syslast;
              quit;
    %end;
```

slide 11

Loops: List Processing

loops: list processing

list contains items sets
and arrays contain elements

an item can be an atom, one variable with information
or a list

a list does not contain data, i.e. summable numeric facts

each row contains a set of parameters

other terms: data- or table-driven, self-modifying, dynamic programs

slide 12

list processing tools

cx-include: call execute a parameterized include

macro callmacro

macro calltext

macro dateloop

slide 13

using macro loops in functions

```
%let macro_name = proc_freq;
%let list       = a bb ccc;
%let n_items    = %sysfunc(countw(&list,%str( )));

%do i = 1 %to &n_items;
    %let item = %scan(&list,&i);
    %put echo: &=item;
    %&macro_name(item=&item)    %* know semicolon!;
    %end;
```

slide 14

Issues when Using Macros

- autocall autoexec: `filename macros'...';`
  `options sasautos = ( macros sasautos);`
- macro functions return tokens
  less than statement
- quoting `%let x = %nrstr(...);`
  `%put %unquote(&x);`
- suffix of reference is dot: `&mvar.` `format&length..`
  `&libname..memname`
  `&filename..txt`

slide 15

Summary

Reasons to use macros Fehd [8]

- variables: program with global symbol table
- control: add statements or branching
- loops: write macro definitions as functions

slide 16

Author Information

Ronald J. Fehd	`Ron.Fehd.macro.maven@gmail.com`
sco.wiki	`http://www.sascommunity.org/wiki/Ronald_J._Fehd`
LinkedIn	`www.linkedin.com/Ronald.Fehd`
affiliation	Stakana Analytics, Senior Maverick
also known as	`macro maven` on SAS-L, Theoretical Programmer

Programs: `http://www.sascommunity.org/wiki/`

Macro_CallMacro Fehd [4]

Macro_CallText Fehd [3]

Macro_DateLoop Fehd [5]

Routine CxInclude Fehd [6]

Conditionally Execute Global Statements Fehd [2]

Trademarks

References

[1] Editor R.J. Fehd. Writing testing aware programs. In *sasCommunity.org*, 2007. URL `http://www.sascommunity.org/wiki/Writing_Testing_Aware_Programs`.

[2] Editor R.J. Fehd. Conditionally executing global statements. In *sasCommunity.org*, 2008. URL `http://www.sascommunity.org/wiki/Conditionally_Executing_Global_Statements`. topics: combining functions sysfunc and ifc; info: example assertions, caveats on logical comparisons.

[3] Editor R.J. Fehd. Macro Call-Macro. In *sasCommunity.org*, 2012. URL `http://www.sascommunity.org/wiki/Macro_CallMacr`. using SCL functions to read a data set and call macros.

[4] Editor R.J. Fehd. Macro Call-Text. In *sasCommunity.org*, 2012. URL `http://www.sascommunity.org/wiki/Macro_CallText`. using SCL functions to read a data set and return tokens within a statement.

[5] Editor R.J. Fehd. Macro loops with dates. In *sasCommunity.org*, 2013. URL `http://www.sascommunity.org/wiki/Macro_Loops_with_Dates`. example macros, programs and updates.

[6] Ronald J. Fehd. List processing routine CallXinc: Calling parameterized include programs using a data set as list of parameters. In *Western Users of SAS Software Annual Conference Proceedings*, 2009. URL `www.lexjansen.com/wuss/2009/app/APP-Fehd2.pdf`. Applications Development, 20 pp.; call execute, data review, data structure, dynamic programming, list processing, parameterized includes, examples.

[7] Ronald J. Fehd. An autoexec companion, allocating location names during startup. In *MidWest SAS Users Group Annual Conference Proceedings*, 2015. URL `http://www.lexjansen.com/mwsug/2015/BB/MWSUG-2015-BB-10.pdf`. Beyond Basics, 15 pp.; autocall macros, global symbol table, filerefs, librefs, cexist catalogs, exist data set, sasautos.

[8] Ronald J. Fehd. Do we need macros? an essay on the theory of application development. In *MidWest SAS Users Group Annual Conference Proceedings*, 2015. URL `http://www.lexjansen.com/mwsug/2015/BB/MWSUG-2015-BB-11.pdf`. Beyond Basics, 10 pp.