

## 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

---

|                      |                               |          |
|----------------------|-------------------------------|----------|
| <b>In this paper</b> | <b>Introduction</b>           | <b>1</b> |
|                      | <b>How SAS works</b>          | <b>3</b> |
|                      | <b>Conditionals</b>           | <b>3</b> |
|                      | <b>Loops: List Processing</b> | <b>4</b> |
|                      | <b>Summary</b>                | <b>5</b> |
|                      | <b>References</b>             | <b>6</b> |

---

---

### Introduction

**Overview**      These slides discuss my overview of programming and software design.

---

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

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

---

slide 2

**aspects of learning  
computer language**

- variables      constants      scope: global or local symbol table
- conditions      if... then... else...      stop  
                          comparisons: *lt, le, eq, ge, gt*      additional code  
                          logical operators: *and, or, not*      branching
- loops      conditional exit      while    leave    until  
                  enumerate: iterative      start, stop, step=n  
                                                  sequence: step eq 1  
                                                  series: step ne 1  
                  itemize: list of items
- functions      macro, method, process, or procedure

slide 3

**HIPO, three aspects of  
every program**

**Hierarchical**

location

type

calls

top

module

routines

middle

routine

subroutines

bottom

subroutine

**Input**

*ingredients*

data structure

**Process**

*algorithm*

process

procedure

N steps

Q: if ... else

executed:

always

conditionally

**Output**

*name of recipe*

data structure

slide 4

**data structure or  
algorithm?**

20%

80%

compiled

executed

data structure

algorithm

object

method

attribute statement(s)

less than 1% of program

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

slide 5

---

## How SAS works

---

### startup process

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

---

slide 6

### autoexec process

adding entries to global symbol table Fehd [11]

|                         |                                         |             |
|-------------------------|-----------------------------------------|-------------|
| 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 7

---

## Conditionals

Conditionally execute global statements using `sysfunc` and `ifc`, [3]

True is not false, [14]

Truth table [2]

Writing testing-aware programs, [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 8

### additional statements

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

---

slide 9

## 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;
```

**Note:** the condition `zero not equal &testing` changes any value to *true*

slide 10

## Loops: List Processing

### loops: list processing

list contains items

sets

$n = \text{cardinality}$

arrays contain elements

$n = \text{dimension}$

an item can be

an attribute, 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

dynamic programs

self-modifying

slide 11

### list processing tools

cx-include: call execute a parameterized include [8]

macro callmacro [10]

macro calltext [13]

macro dateloop [9]

slide 12

### using macro loops in functions

```
%let macro_name = proc_freq;
%let list       = a bb ccc;
                *count word(s) delimited by space;
%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;
```

**Note:** \*count delimiters and add one;

```
%let n_items = %eval(%sysfunc(countc(&list,%str( )))+1);
```

Fehd and Carpenter, List Processing Basics, 2007 [15]

slide 13

## 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 14

## reasons to use macros

- variables: programs have access to global symbol table
- control: add statements or branching
- loops: write macro definitions as functions

**Notes:** Do we need macros? [12]  
Macro design ideas [7]

---

slide 15

---

## Summary

### Conclusion

Using SAS software effectively requires gaining an understanding of software design issues.

---

## Author Information

Ronald J. Fehd [Ron.Fehd.macro.maven@gmail.com](mailto:Ron.Fehd.macro.maven@gmail.com)  
sco.wiki [http://www.sascommunity.org/wiki/Ronald\\_J.\\_Fehd](http://www.sascommunity.org/wiki/Ronald_J._Fehd)  
LinkedIn [www.linkedin.com/Ronald.Fehd](http://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](http://www.sascommunity.org/wiki/Macro_CallMacro) [5]  
[Macro\\_CallText](http://www.sascommunity.org/wiki/Macro_CallText) [4]  
[Macro\\_DateLoop](http://www.sascommunity.org/wiki/Macro_DateLoop) [6]  
[Routine\\_Call-Exec-Include](http://www.sascommunity.org/wiki/Routine_Call-Exec-Include) [8]  
[Conditionally Execute Global Statements](http://www.sascommunity.org/wiki/Conditionally_Execute_Global_Statements) [3]  
[List Processing Basics](http://www.sascommunity.org/wiki/List_Processing_Basics) Fehd and Carpenter [15]

---

## References

- [1] Editor R.J. Fehd. Writing testing aware programs. In *sasCommunity.org*, 2007. URL [http://www.sascommunity.org/wiki/Writing\\_Testing\\_Aware\\_Programs](http://www.sascommunity.org/wiki/Writing_Testing_Aware_Programs).
  - [2] Editor R.J. Fehd. Truth table of logical operators *and*, *or*, *not*, *xor* and De Morgan's laws of *nand*, *nor*. In *sasCommunity.org*, 2007. URL [http://www.sascommunity.org/wiki/Truth\\_Table](http://www.sascommunity.org/wiki/Truth_Table).
  - [3] Editor R.J. Fehd. Conditionally executing global statements. In *sasCommunity.org*, 2008. URL [http://www.sascommunity.org/wiki/Conditionally\\_Executing\\_Global\\_Statements](http://www.sascommunity.org/wiki/Conditionally_Executing_Global_Statements). topics: combining functions sysfunc and ifc; info: example assertions, caveats on logical comparisons.
  - [4] Editor R.J. Fehd. Macro Call-Macro. In *sasCommunity.org*, 2012. URL [http://www.sascommunity.org/wiki/Macro\\_CallMacr](http://www.sascommunity.org/wiki/Macro_CallMacr). using SCL functions to read a data set and call macros.
  - [5] Editor R.J. Fehd. Macro Call-Text. In *sasCommunity.org*, 2012. URL [http://www.sascommunity.org/wiki/Macro\\_CallText](http://www.sascommunity.org/wiki/Macro_CallText). using SCL functions to read a data set and return tokens within a statement.
  - [6] Editor R.J. Fehd. Macro loops with dates. In *sasCommunity.org*, 2013. URL [http://www.sascommunity.org/wiki/Macro\\_Loops\\_with\\_Dates](http://www.sascommunity.org/wiki/Macro_Loops_with_Dates). example macros, programs and updates.
  - [7] Editor R.J. Fehd. Macro design ideas. In *sasCommunity.org*, 2014. URL [http://www.sascommunity.org/wiki/Macro-Design\\_Ideas](http://www.sascommunity.org/wiki/Macro-Design_Ideas). history, theory, templates for documentation and macros, best practices.
  - [8] 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](http://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.
  - [9] Ronald J. Fehd. Writing macro do loops with dates from then to when. In *MidWest SAS Users Group Annual Conference Proceedings*, 2013. URL <http://analytics.ncsu.edu/sesug/2013/CC-03.pdf>. 20 pp.; topics: dates are integers, formats and functions to convert date references to integers, calculations, bibliography.
  - [10] Ronald J. Fehd. List processing macro call-macro. In *MidWest SAS Users Group Annual Conference Proceedings*, 2014. URL [www.mwsug.org/proceedings/2014/BB/MWSUG-2014-BB04.pdf](http://www.mwsug.org/proceedings/2014/BB/MWSUG-2014-BB04.pdf). Coders Corner, 19 pp.; using %sysfunc with SCL functions to read a list, a control data set, and for each observation, call a macro with variable names and values as named parameters.
  - [11] 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.
  - [12] 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.
  - [13] Ronald J. Fehd. List processing macro call-text. In *MidWest SAS Users Group Annual Conference Proceedings*, 2016. Tools of Trade, 10 pp.; using %sysfunc with SCL functions to read a list, a control data set, and for each observation, return %unquoted text.
  - [14] Ronald J. Fehd. True is not false: Evaluating logical expressions. In *SouthEast SAS Users Group Conference Proceedings*, 2016. URL <http://www.lexjansen.com/sesug/2016/BB-186.pdf>. Beyond the Basics, 9 pp.; Boolean algebra, Boolean logic, De Morgan's law, evaluation, logical operators, sql joins.
  - [15] Ronald J. Fehd and Art Carpenter. List processing basics: Creating and using lists of macro variables. In *SAS Global Forum Annual Conference Proceedings*, 2007. URL <http://www2.sas.com/proceedings/forum2007/113-2007.pdf>. Hands On Workshop, 20 pp.; comparison of methods: making and iterating macro arrays, scanning macro variable, writing calls to macro variable, write to file then include, call execute; using macro function nrstr in call execute argument; 11 examples, bibliography.
-