

The Best of Cheesy, Sleazy SAS® Tricks

Presented at NESUG 2003
By Michael Davis
Bassett Consulting Services, Inc.

Introduction

- Paper based on off-hand comment
- Tips solicited on SAS-L
- What is "Cheesy" or "Sleazy" ?
 - Coding declarations that suppress over-sollicitous features of SAS software
 - Tricks that hint at unexpected SAS behavior

Add Dummy Params to Macros

- What if an addt'l parameter needed after a macro has been deployed?
- Code as

```
%macro noparms (dummy=) ;
```
- Call as

```
%noparms ( ) ;
```

Do It In Three (or Four)

- Boolean test used with MERGE statement
 - `if a=1 and b=1`
 - `if a and b`
 - `if a*b`
 - `if a&b;`
- Ampersand represents logical AND

This Variable Is Uninitialized

- NOTE: Variable z is uninitialized

```
data ... ;  
  /* prevent NOTE: Variable ***** is  
  uninitialized messages ;  
  retain _character_ ' ' _numeric_ . ;  
  stop ;  
run ;
```

ERR-OR, WARN-ING

- Goal is to prevent "ERROR" or "WARNING" from appearing in log

```
data _null_ ;  
  put "WARN" "ING: -bad thing!  
  Check!" _all_ ;  
run ;
```

- Note the absence of a concatenation operator

Tweaking PROC FREQ SORT ORDER

- Report comes out in format label order
- Use leading blank to tweak the order

```
value myfmt  
1=" High School"  
2="College"  
3="Grad School" ;
```

Global Macro Declaration Trick

- Has global macro variable been set ?
 - `%put The Length of NESUG is %length(&nesug) ;`
 - **WARNING:** Apparent symbolic reference NESUG not resolved.
- Solution: `%global nesug ;`
- Log: **The Length of NESUG is 0**

Left Side Functions

- Left side of an assignment can be altered !

```
substr(name, 2) =  
    lowercase(substr(name, 2)) ;
```

- In this example, "RALPH" becomes "Ralph"

Quick Way to Read Mixed-Case Text

```
data xyz ;  
    infile 'mydata.txt' ;  
    input @ ;  
    _infile_ = uppercase(_infile_) ;  
    input ... ;  
run ;
```

- Why sleazy? Modification of `_infile_`

Debug PROC REPORT COMPUTE Blocks

- Put statement yields the following error:

```
ERROR: PUT statement is not valid  
in this context.
```

- Insert the following in the block:
`call execute('%put' || <variable>) ;`
- Why sleazy? CALL EXECUTE usually references a macro, not a %PUT

Conclusion

- Illustrate the subtle nature of SAS
- Features are well embedded in SAS and should be available in subsequent releases and across platforms
- Author may revisit this subject in the future so keep those "cheesy" and "sleazy" tricks coming in



Contact Information

Michael Davis
Bassett Consulting Services, Inc.
10 Pleasant Drive
North Haven CT 06473-3712
Email: michael@bassettconsulting.com
Web: <http://www.bassettconsulting.com>