



Building Reusable Programs As Includes Or Macros

Ronald J. Fehd

October 21, 2005

Outline

Theory

- Ideas

- Guidelines

- Documentation

- Polishing

Practice

- Find Code

- Test as Include

- Make Macro

- Comparison

- Q and A



Music: KSAs

- ▶ pattern recognition
- ▶ sequential processing
- ▶ cymbal manipulation



Reuse is Reduction

- ▶ many to few
- ▶ less is more
- ▶ hide complexity
- ▶ encapsulation



Steps in Building Reusable Code

- ▶ Find Example
- ▶ Polish Statements
- ▶ Test as Include
- ▶ Test as Macro
- ▶ Polish Documentation



Style Sheet

My Style

summer: shorts
t-shirt
sandals

fall: jeans
long-sleeves
Birkenstocks

...



Style Sheet

XYZ Company SAS Style Sheet

Header

Code: Capitalization
 Naming Conventions
 White Space

Test Data



Program Header

- ▶ Requirements
- ▶ Context
- ▶ Specifications
- ▶ Usage



Programmer Specifications

- ▶ Input
- ▶ Process
- ▶ Output



Naming Conventions, from Recipes

- ▶ Chocolate Cake
- ▶ Ingredients
- ▶ Preparation
- ▶ Output
- ▶ Input
- ▶ Process



Program Header List

- ▶ Name:
- ▶ Requirements:
- ▶ Contexts:
- ▶ Specifications:
- ▶ Usage:



Program Header, details

Name: FileName.sas

Requirements: description:
 purpose:

Contexts:

Specifications:

Usage:



Program Header, details

Name:

Requirements:

Contexts: program group:

 program type:

 SAS type:

Specifications:

Usage:



Program Header, details

Name:

Requirements:

Contexts:

Specifications: input:
 process:
 output:

Usage:



Capitalization Styles

- ▶ UPPER CASE: ALL CAPS
- ▶ lower case



Capitalization Styles

- ▶ UPPER CASE: ALL CAPS
- ▶ lower case
- ▶ Initial Caps
- ▶ InternalCaps



Adding White Space

- ▶ per line
- ▶ using tab key



Adding White Space

- ▶ per line
- ▶ using tab key
- ▶ gestalt
- ▶ use to group similar tokens



7 Habits of Highly Effective People

Begin
With the End
in Mind

Steven Covey



Export1fromWizard.sas

```
PROC EXPORT DATA= SASHELP.CLASS
              OUTFILE= "SASHELP.XLS"
              DBMS=EXCEL REPLACE;
              SHEET="CLASS";
RUN;
```

Export2polished.sas

```
*          keyword  values
-----  -----  -----  =  -----;
PROC Export data    =  SAShelp.Class
          outFile = "SAShelp.xls"
          dbms     =  Excel
          replace          ;
          sheet  = "Class";

run;
```

Export3parameters.sas

```
%Let In_Lib          = SAShelp;
%Let In_Data         = Class;
%Let Out_File        = SAShelp;
%Let Out_Sheet       = Class;

PROC Export data      =      &In_Lib.&In_Data.
                   outFile = "&Out_File.xls"
                   dbms    = Excel
                   replace          ;
                   sheet   = "&Out_Sheet";

run;
```

Export2xlsInc.sas

```
/*      name: Export2xlsInc.sas
description: Export data to Excel
      usage:
%Let In_Lib      = SAShelp;
%Let In_Data     = Class;
%Let Out_File    = SAShelp;
%Let Out_Sheet  = Class;
%Inc Project(Export2xlsInc);
**** ..... */
PROC Export data      =      &In_Lib.&In_Data.
      outFile = "&Out_File..xls"
      dbms      = Excel
      replace ;
      sheet    = "&Out_Sheet.";

run;
```



Export2xlsIncTest.sas

```
/*      name: Export2xlsIncTest.sas
description: Test of Export as include
**** ..... */
options source2;
%let In_Lib      = SAShelp;
%let In_Data    = Class;
%let Out_File   = SAShelp;
%let Out_Sheet  = Class;
%inc Project(ExportInc);

%let In_Data    = CitiYr;
%let Out_Sheet  = &In_Data.;
%inc Project(ExportInc);
```



Export2xls.sas

```
/*      name: Export2xls.sas
```

```
Requirements: -----
```

```
  description: convert data set to Excel
```

```
  purpose: write one sheet in Excel file
```

```
Contexts: -----
```

```
  program group: conversion
```

```
  program type: subroutine
```

```
  SAS type: macro, with parameters
```

```
Specifications: -----
```

```
  input: libref, data set name
```

```
  process: export, replace
```

```
  output: Excel filename, sheet
```



Export2xls.sas, continued

```
%Macro Export2xls
    (In_Data    =
    ,In_Lib     = Work    /* default */
    ,Out_File   =
    ,Out_Sheet  = Sheet1 /* default */
    )/ des      = 'site: Export to Excel sheet';

PROC Export data    =    &In_Lib.&In_Data.
    outfile = "&Out_File..xls"
    dbms    = Excel
    replace ;
    sheet   = "&Out_Sheet.";

run;%Mend;
```

Comparison of:

| | Includes | Macros |
|------------------------|----------|--------|
| use global macro vars | Y | N |
| loops | N | Y |
| conditional execution | N | Y |
| set parameter defaults | N | Y |
| test parameter values | N | Y |
| program types: | | |
| routine, subroutine | Y | Y |
| function | N | Y |
| debugging, testing | | |
| option to echo | source2 | mprint |
| line numbers | Y | N |



Summary

- ▶ Building for Reuse requires many steps
 - ▶ Find, Test, Polish, ... Document
- ▶ Documentation is the most important

About The Speaker

Speaker: Ronald Fehd
SAS-L's macro maven

Company: Centers for Disease Control
somewhere in
Atlanta, GA, USA

Phone: 770-488-8102

E-mail: RJF2@cdc.gov