PROC FCMP
Function CoMPiler

March 18, 2014

Boston Area SAS Users Group

Robert Rosofsky
Health Information Systems Consulting LLC

What is the requirement?

- Need to create single assignment statements

  Variable1 = some function of (Var2, Var3, VarN)
- Multiple statements not permitted; for example:
  - IF-THEN-ELSE
  - SELECT-WHEN-END
  - Other complicated logic
  - Arrays
- Return a single value only

PROC FCMP……*to the rescue*

- Enables development of user-defined functions
- Uses DATA-step like code, including:
  - Assignment statements
  - SAS functions
  - Arrays
  - IF – THEN – ELSE, DO loops, etc.
- Functions available to the entire session/program
- Permitted and Useful in
  - DATA step
  - Some statistical procedures
  - SQL

Basic Structure of PROC FCMP

```
PROC FCMP /* options to be discussed later */ ;

    function NameOfFunction (Parm1, Parm2, ParmN);
        lots of code goes here
    endsub;

    subroutine NameOfSubroutine (Parm1, Parm2, ParmN);
        lots of code goes here
    endsub;

RUN;
```

Parameters for PROC FCMP

Passed parameters can be:

- Numeric (Num1, Num2, NumN)
- Character (Char1 **$**, Char2 **$**, CharN **$**)
- *Mix it up* (Num1, Char1 **$**, Num2)

Returned Value from PROC FCMP

Returned value can be:

- Numeric*:* Function *NameOfFunction* (*....parms...*) ;
- Character: Function *NameOfFunction* (*....parms...*) **$ length**;

Example #1

Validate a SAS name (i.e., dataset name or variable name)

Rules:

- Length: 32 characters or less
- Start: Underscore or letters
- Contains: Underscores, letters, or digits

Example #1

```
function SASValidName (NM $)  ;
    name = left(trim(nm));
    /* Length 32 characters or less */
    if length(name) > 32
        /* Nothing remains after dropping letters, digits & underscores */
        or lengthn(compress(name,"","adf")) > 0
        /* First character only: nothing remains after dropping letter &
        underscore */
        or lengthn(compress(substr(name,1,1),"","af")) > 0
        /* No embedded blanks*/
        or length(compress(name)) NE length(strip(name))
        then return(0);
        else return(1);
 endsub;
```

← Note that numerics are returned, even though the value checked is character

Example #1
DATA Step

```
data _null_;
    SAS_Name1 = “Is_This_A_SAS_Name1”;
    SAS_Name2 = “_Is_This_A_SAS_Name2”;
    SAS_Name3 = “_Is_This_A_SAS_Name3?”;

    NameValid1 = SASValidName(SAS_Name1);
    NameValid2 = SASValidName(SAS_Name2);
    NameValid3 = SASValidName(SAS_Name3);
    put NameValid1= NameValid2= NameValid3= ;
run;
```

NameValid1=1 NameValid2=1 NameValid3=0

Example #2

Determine if a concatenated series of character values is valid or not

Rules:

- Passed value must contain only valid values

Example:

- String can contain only these values: AA, BB, 01, 02, 03
- If String="AA0201", it is valid
- If String="BB01CC", it is *not* valid

Example #2

```
function ValQCChar(StringVar $, SearchValues $);
    ValSize=length(scan(SearchValues,1,"^"));
    if mod(length(trim(StringVarIn)),ValSize) NE 0 then Valid=0;
    else do;
        Valid=1;
        do chunk=1 to (length(trim(StringVar)) / ValSize) ;
            if index(SearchValues,substr(StringVar, chunk*ValSize -(ValSize-1),
            ValSize)) = 0 then Valid=0;
        end;
    end;
    *** Return value ;        return(Valid);
endsub;
```

Sample call: Valid= ValQCChar(VarName,”AA^BB^01^02^03”)

Other Examples

function DeliveryMethod_GA(Var1, Var2, Var3, Var4, Var5, Var6, MetaData $) $2;

Given 6 separate variables to indicate a method of birth delivery, return only one value, based on a hierarchy

function ICD10QC(Value $, Missing);

Check that a string meets the pattern for an ICD10 diagnosis code, returning 0 or 1

function ValQCNum(TestVar, LowVal, HighVal, SkipVals $);

Check if numeric values are in a range, but not equal to selected values within the “SkipVals” range;

Some DATA step and FCMP Code Comparisons

DATA Step	PROC FCMP
If name="YES" then X=1; else X = 0;	X = If name="YES" then 1 else 0;
Put x= +5 y= @30	Put (x/y) (ArrName[t])
ArrName[t] or ArrName(t) or ArrName{t}	ArrName[t] or ArrName{t} only

PROC FCMP Options

- Functions are not available simply by running PROC FCMP
- OUTLIB = is the ***one*** required option to make the functions available for use
  - Can be stored in any library (i.e., WORK or permanent)
  - Is a three level name

- Example (for temporary session only)
  - outlib = WORK.FCMP_Functions.one;
- Example (for permanent storage)
  - outlib = PERM.FCMP_Functions.one;

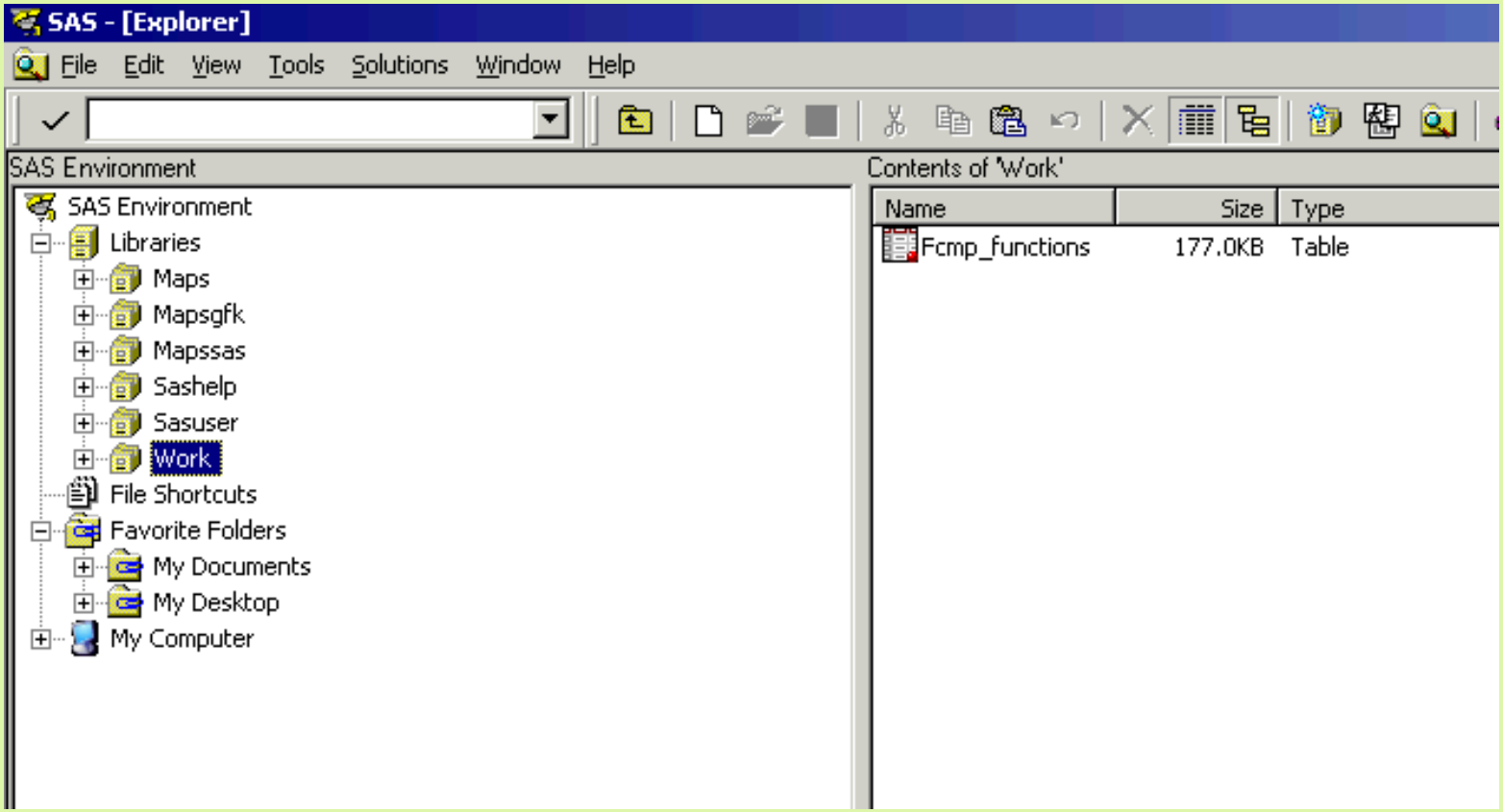
But wait, there's more

- You also need an OPTIONS statement

OPTIONS cmplib = WORK.FCMP_Functions;

  - Note that this is a 2-level name

How are functions stored?



Stored as a dataset

How are functions stored?

SAS - [VIEWTABLE: Work.Fcmp_functions]

File Edit View Tools Data Solutions Window Help

	Key	Owner	Sequence	Type	Subtype	Name
663	F.ONE.MULTIVARS	CMP	661	Statement Source	Executable	ASSIGN
664	F.ONE.MULTIVARS	CMP	662	Statement Source	Comment	CMT
665	F.ONE.MULTIVARS	CMP	663	Statement Source	Executable	IF
666	F.ONE.MULTIVARS	CMP	664	Statement Source	Executable	ASSIGN
667	F.ONE.MULTIVARS	CMP	665	Statement Source	Executable	END
668	F.ONE.MULTIVARS	CMP	666	Statement Source	Comment	CMT
669	F.ONE.MULTIVARS	CMP	667	Statement Source	Executable	IF
670	F.ONE.MULTIVARS	CMP	668	Statement Source	Executable	RETURN
671	F.ONE.MULTIVARS	CMP	669	Statement Source	Executable	ELSE
672	F.ONE.MULTIVARS	CMP	670	Statement Source	Executable	RETURN
673	F.ONE.MULTIVARS	CMP	671	Statement Source	Executable	END
674	F.ONE.MULTIVARS	CMP	672	Statement Source	Executable	ENDSUB
675	F.ONE.SASVALIDNAME	CMP	673	Prototype	FCmp	one
676	F.ONE.SASVALIDNAME	CMP	674	Header	Function	
677	F.ONE.SASVALIDNAME	CMP	675	Statement Source	Executable	FUNCTION
678	F.ONE.SASVALIDNAME	CMP	676	Statement Source	Comment	CMT
679	F.ONE.SASVALIDNAME	CMP	677	Statement Source	Executable	ASSIGN
680	F.ONE.SASVALIDNAME	CMP	678	Statement Source	Executable	ASSIGN
681	F.ONE.SASVALIDNAME	CMP	679	Statement Source	Comment	CMT
682	F.ONE.SASVALIDNAME	CMP	680	Statement Source	Comment	CMT
683	F.ONE.SASVALIDNAME	CMP	681	Statement Source	Executable	IF
684	F.ONE.SASVALIDNAME	CMP	682	Statement Source	Executable	RETURN
685	F.ONE.SASVALIDNAME	CMP	683	Statement Source	Executable	ELSE
686	F.ONE.SASVALIDNAME	CMP	684	Statement Source	Executable	RETURN
687	F.ONE.SASVALIDNAME	CMP	685	Statement Source	Executable	ENDSUB
688	F.ONE.SASVALIDNAME	CMP	686	Symbol		name

How are functions stored?

SAS - [VIEWTABLE: Work.Fcmp_functions]

File Edit View Tools Data Solutions Window Help

	Key	Owner	Sequence	Encoded	Value
675	F.ONE.SASVALIDNAME	CMP	673	.	<L n="Prototype"><S n="Name"><![CDATA[SASValidName]]> n="Group"><![CDATA[]]></S><N n="MaxLag">0</N><N n="Flag0">0</N><N n="Flag1">128</N><S n="ReturnType"><![CDATA[n]]></S><N n="ReturnSize">8</N><L n="ArgList"><L n="Arg"><S n="Name"><![CDATA[NM]]></S><S n="Kind"><![CDATA[v]]></S><S n="Type"><![CDATA[c]]></S><S n="Class"><![CDATA[a]]></S><N n="Status">0</N><N n="Status2">0</N><N n="NInit">0</N><N n="MaxLag">0</N><N n="Size">-33</N><N n="Flag1">192</N><N n="Flag2">0</N><N n="Flag3">0</N><N n="Flag4">0</N><N n="Flag5">64</N><N n="Flag6">32</N></L></L></L>
677	F.ONE.SASVALIDNAME	CMP	675	.	function SASValidName(NM $);
678	F.ONE.SASVALIDNAME	CMP	676	.	* Initialize as valid SAS valid name;
679	F.ONE.SASVALIDNAME	CMP	677	.	true=1;
680	F.ONE.SASVALIDNAME	CMP	678	.	name = left(trim(nm));
681	F.ONE.SASVALIDNAME	CMP	679	.	* Check the following: * - length <= 32 characters ;
682	F.ONE.SASVALIDNAME	CMP	680	.	* - characters only alpha, digits, or underscores * - 1st character only alpha or underscores;

%sysfunc

- Can be used with all user-defined functions created by PROC FCMP
- But this does not work with SAS v9.2, contrary to documentation
- Works only in SAS 9.3+

Example: Macro Use

```
%let SAS_Name1 = Is_This_A_SAS_Name1;
%let SAS_Name2 = _Is_This_A_SAS_Name2;
%let SAS_Name3 = _Is_This_A_SAS_Name3?;

%let NameValid1 = %sysfunc(SASValidName(&SAS_Name1.));
%let NameValid2 = %sysfunc(SASValidName(&SAS_Name2.));
%let NameValid2 = %sysfunc(SASValidName(&SAS_Name3.));

%put &NameValid1.  &NameValid2.  &NameValid3.  ;
```

```
1   1  0
```

www.HealthInfoSys.net