

Use Macro Variables to write SAS Scripts (Beginner level)

By Sylvia So

Research & Data Analyst

Data Access,

Customer Relationship Management & Data Access Unit
(CRMDA)

Table of Contents

1. Background
2. What is the Issue?
3. Option 1 – Call Execute
4. Option 2 – PUT
5. Questions & Comments

Background



Submit a data request to:

<http://www.health.alberta.ca/initiatives/health-research.html>

Data Extract Parameter
Name
DOB
PHN
Type of investigation
Time Period
Diagnosis (ICD) Code
Other etc ...

Main Macro


The screenshot shows a software editor window titled "Editor - Untitled1". The main content is a macro definition: `%macro My_Report (Request_No, Name, DOB, PHN, etc...);`. Below this, a grey box contains a list of steps: `%Macro My_report();`, followed by three sets of actions: "Pull data1", "Analyze", "Report findings1"; "Pull data2", "Analyze", "Report findings2"; and "Pull data3", "Analyze", "Report findings3". The list ends with "... .." and `%MEND;`. A large green arrow points from this list to a 3D illustration of a green folder containing a spreadsheet with columns A, B, and C, and a bar chart with three bars of increasing height.

Main Macro

```
Editor - Untitled1

+ %macro My_Report (Request_No, Name, DOB, PHN, etc...);

%My_Report (
  Request_No=REQ-00001,
  Type=Investigation,
  First_Name=Bingo,
  Last_Name=Lee,
  PHN=123456789,
  DOB=01Jan1960,
  Periods=01jan2017:31dec2017,
  Excel_File=My_00001 - 100100.1.1.xlsx
);
```



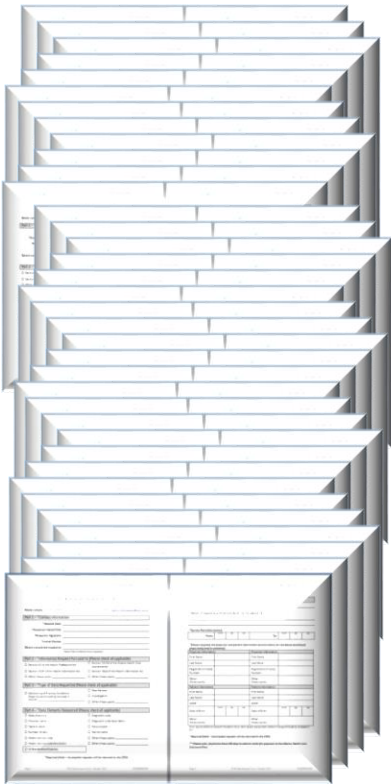
Multiple Requests



Wouldn't it be nice if....



Option 1 – Driver Table



Request_no	FirstName	LastName	PHN	DOB	Etc.
00001	Bingo	Lee	123456789	01Jan1960	...
00002	Jaka	So	101010101	02Feb1990	...
00003
00004
...

Option 1 – CALL EXECUTE

```

Editor - Untitled1
|

*Apply the "CALL EXECUTE" function;
data _null_;
  Set work.driver_table;
  words=
    '%My_Report(...=' || trim(FirstName) || ',...=' || trim(LastName) || ',....)';
  CALL EXECUTE(%nrstr (words));
run;

```

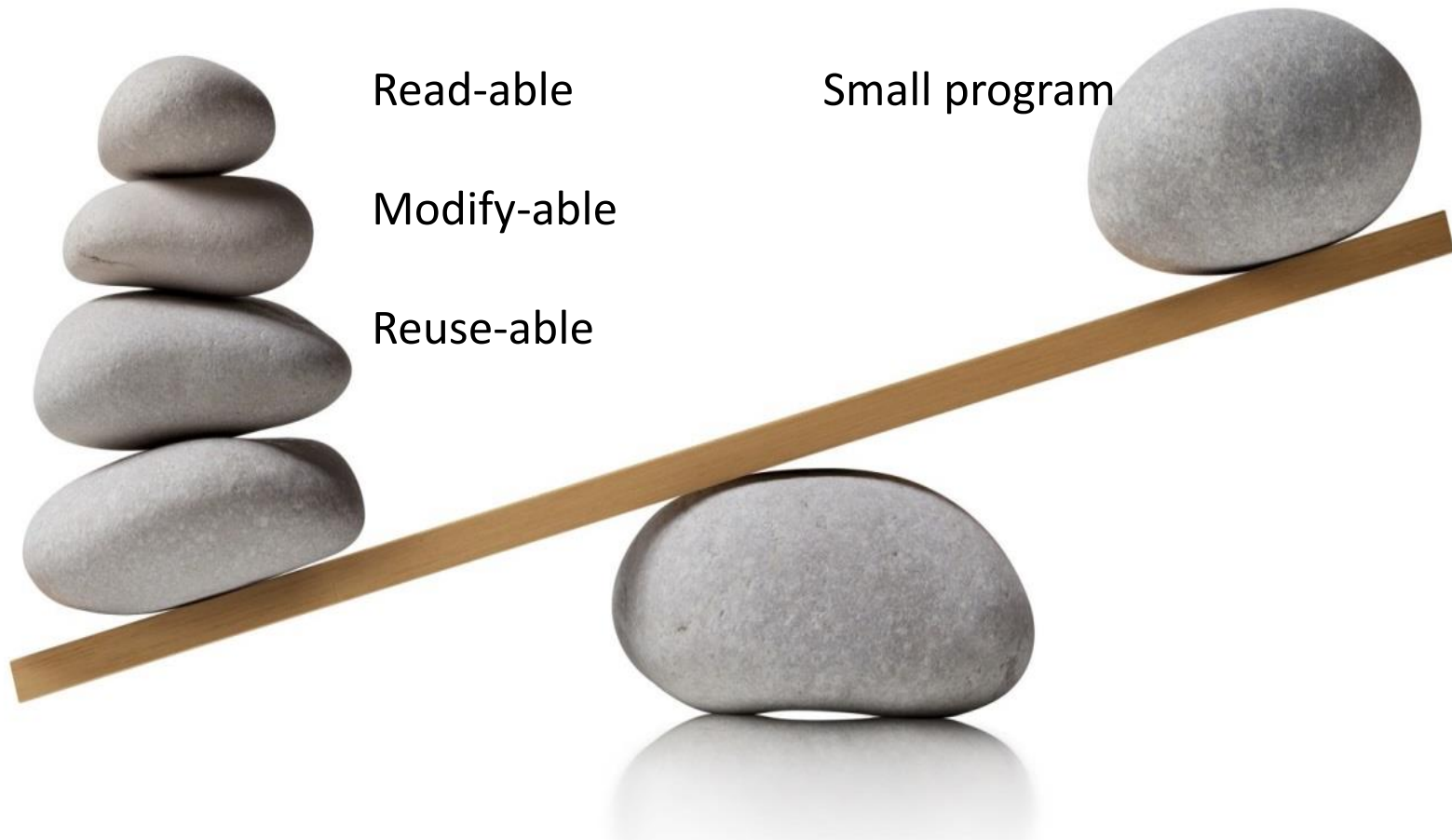
Option 1 – SAS log

```

Log - (Untitled)
1  + %My_Report(REQ_no=00001, FirstName=Bingo,
   LastName=Lee, PHN=123456789, DOB=01Jan1960,
   .....);
2  + %My_Report(REQ_no=00002, FirstName=Jaka,
   LastName=So, PHN=101010101, DOB=02Feb1990,
   .....);
3  + %My_Report(REQ_no=00003, FirstName=xxx,
   LastName=xxx, PHN=xxxxxxxxxxx, DOB=xxxxxxxxx, .....);
4  + .....
5  + ....
6  + ...

```

A Balancing Act



... leave a trail



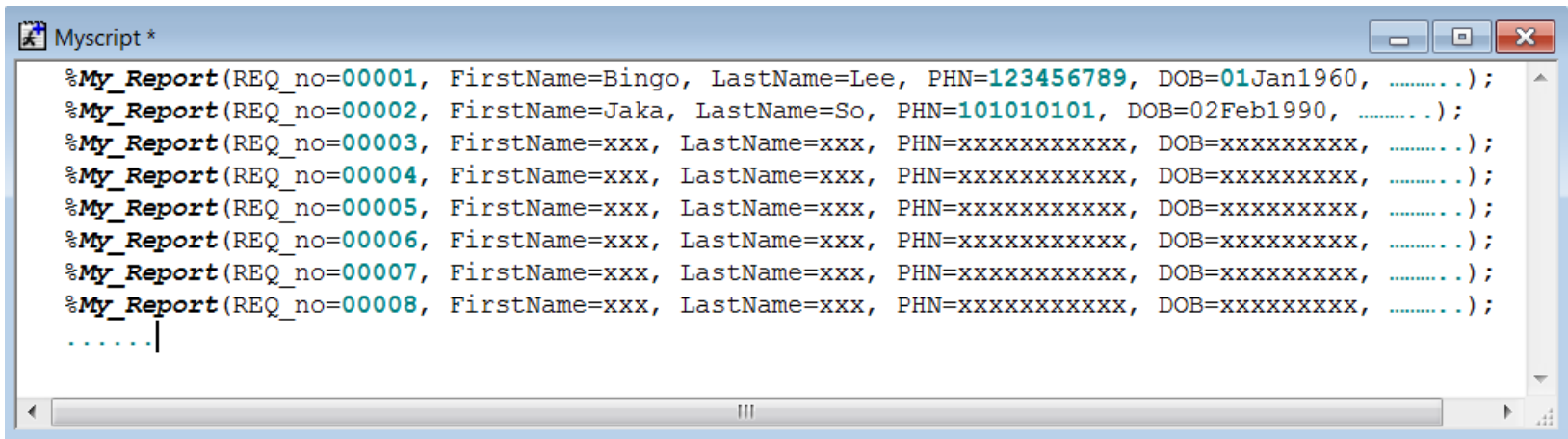
Option 2 – PUT

```

Editor - Untitled1
|
*Set up location for MyScript and PUT [Words into it;
filename Myscript "$path\Myscript.sas";
data _null_;
  Set work.driver_table;
  file Myscript;
  words=
    '%My_Report(...=' || trim(FirstName) || ',...=' || trim(LastName) || ',....)';
  PUT Words;
run;

```

Option 2 – log.txt



```
%My_Report(REQ_no=00001, FirstName=Bingo, LastName=Lee, PHN=123456789, DOB=01Jan1960, .....);  
%My_Report(REQ_no=00002, FirstName=Jaka, LastName=So, PHN=101010101, DOB=02Feb1990, .....);  
%My_Report(REQ_no=00003, FirstName=xxx, LastName=xxx, PHN=xxxxxxxxxxxx, DOB=xxxxxxxx, .....);  
%My_Report(REQ_no=00004, FirstName=xxx, LastName=xxx, PHN=xxxxxxxxxxxx, DOB=xxxxxxxx, .....);  
%My_Report(REQ_no=00005, FirstName=xxx, LastName=xxx, PHN=xxxxxxxxxxxx, DOB=xxxxxxxx, .....);  
%My_Report(REQ_no=00006, FirstName=xxx, LastName=xxx, PHN=xxxxxxxxxxxx, DOB=xxxxxxxx, .....);  
%My_Report(REQ_no=00007, FirstName=xxx, LastName=xxx, PHN=xxxxxxxxxxxx, DOB=xxxxxxxx, .....);  
%My_Report(REQ_no=00008, FirstName=xxx, LastName=xxx, PHN=xxxxxxxxxxxx, DOB=xxxxxxxx, .....);  
.....|
```

PUT vs CALL EXECUTE

```

*Scripting with PUT;
filename Myscript 'Myscript.sas';

data _null_;
  Set work.driver_table;
  file Myscript;

words= '%My_Report(...)' ;

PUT Words;
  run;

```

```

*Routine "CALL EXECUTE";

data _null_;
  Set work.driver_table;

words= '%My_Report(...)' ;

CALL EXECUTE(%nrstr(words));
  run;

```




Reference

Robert Williams, Paper CC104, Let SAS® Do the Coding for You!

<http://analytics.ncsu.edu/sesug/2014/CC-12.pdf>

Leonid Batkhan, SAS Blogs. CALL EXECUTE made easy for SAS data-driven programming

<https://blogs.sas.com/content/sgf/2017/08/02/call-execute-for-sas-data-driven-programming>