

Disclaimer

I'm not a native speaker but I'm doing my best.

If I say something that may be interpreted as either negative or positive (e.g. a joke vs. an offense) I always, always, mean the positive.

Macro Variable Arrays Made Easy with macroArray **SAS** Package

AP-108



Bartosz Jabłoński

✉ yabwon@gmail.com
[linkedin.com/in/yabwon](https://www.linkedin.com/in/yabwon)

May 19th - 22nd, 2024
PharmaSUG
Baltimore



Takeda
Pharmaceutical

www.takeda.com



Experis

Manpower Group

www.experis.com



www.polsug.com
#PolSUG

mva



Macro Variable Arrays

A **Macro Variable Array** (also, MVA or macro array) is a jargon term for a list of macro variables with a common prefix and numerical sequential suffixes.

```
_____ code: simple MVA _____  
1  %let a1 = Macro;  
2  %let a2 = Variables;  
3  %let a3 = Array;
```

Usually to get values of an MVA a technique called indirect referencing is used.

`&&prefix&suffix`

Indirect Reference

- double ampersands (&&) are replaced with a single one (&),
- macro variables (&macVar) are resolved to the value (value),
- any non-macro text is kept "as is",
- repeat until there are no more macrotriggers.

code: call to MVA value

```
1 %let a1 = Macro;  
2  
3 %let i=1;  
4  
5 %put &&a&i;
```

the log - indirect referencing

```
1 &&a&i  
2  
3 && a &i  
4  
5 & a 1  
6  
7 &a1  
8  
9 Macro
```

"Standing on the shoulders of giants"

- Ted Clay
- David Katz

```
_____ code: %array() _____  
1 %ARRAY(days, VALUES=Monday Wednesday Friday Sunday)  
2 %put _user_;
```

```
_____ the log - result _____  
1 global days1 Monday  
2 global days2 Wednesday  
3 global days3 Friday  
4 global days4 Sunday  
5 global daysN 4
```

"Standing on the shoulders of giants"

- Ted Clay
- David Katz

```
code: %do_over()  
1 data Want;  
2   set Have;  
3   where dayName in ( %DO_OVER(days, PHRASE="?"));  
4   total + value;  
5 run;
```

```
the log - result  
1 data Want;  
2   set Have;  
3   where dayName in ("Monday" "Wednesday" "Friday" "Sunday");  
4   total + value;  
5 run;
```

"Standing on the shoulders of giants"

- Ted Clay
- David Katz

- ① "This is brilliant!"
- ② "How does it work?!"
- ③ "Can its functionality be improved?"

January 18, 2019

- ① Can we have %array() working as "pure macro code" (even for creating macro arrays from data sets)?
- ② MVAs are "one based". Can an MVA created by %array() start indexing from an arbitrary non-negative integer (e.g., 0, 2, 3, etc.)?
- ③ We can create multiple MVAs from multiple variables from a data set. Can we also create MVAs with lists of unique values?
- ④ Can we make indirect referencing call to the macro array values more "convenient", e.g., similar to the data step array, like: (prefix[suffix])?
- ⑤ Can we use SAS functions to alter, modify, or generate MVA values?
- ⑥ Can we have syntax similar to the one known from 4GL to create MVAs?
- ⑦ Can all those extension be added keeping backward compatibility?

All the answers are: yes we can!

And now, our little internal "Bob the Builder" wants details.

macroArray



The macroArray SAS Package

At the moment the package contains the following macros:

- | | |
|----------------------|----------------------|
| ① %appendArray(), | ⑨ %do_over3(), |
| ② %appendCell(), | ⑩ %make_do_over(), |
| ③ %array(), | ⑪ %mcHashTable(), |
| ④ %concatArrays(), | ⑫ %mcDictionary(), |
| ⑤ %deleteMacArray(), | ⑬ %QzipArrays(), |
| ⑥ %do_over(), | ⑭ %zipArrays(), and |
| ⑦ %do_over2(), | ⑮ %sortMacroArray(). |

links



SAS Packages Framework

„The wolf is sated and the sheep is whole”



sas
packages

the way to share

github.com/yabwon/SAS_PACKAGES

by Bart Jablonski

ORLY[®]

macroArray in SASPAC

„The wolf is sated and the sheep is whole“



sas
packages

the way to share

github.com/SASPAC/macroarray

by Bart Jablonski

ORLY⁷

Workshop Materials



github.com/yabwon/How-SASPackages

demo



code used for the demo



github.com/SASPAC/macroarray/extras

dziękuję

