

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.

A Gentle Introduction to SAS Packages

HT-111



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

Technical Requirements

The workshop assumes that you have available the following resources:

- ① a computer with internet access,
- ② a SAS session (may be local or remote), the interface can be: SAS DMS, SAS EG, or SAS Studio, can be SAS 9.4 or Viya with SPRE
- ③ the SAS session has:
 - direct internet access or
 - you are able to upload files to the computer hosting your session,
 - (nice to have but not required) the SAS session has XCMD option enabled.

Additional assumption is that **you wrote at least a few of those**: a macro, a function (fcmp, proto, or casl), a format or informat, or an IML module.

Inspirations

- Comprehensive T_EX Archive Network
<https://ctan.org/pkg/>
- Comprehensive R Archive Network
<https://cran.r-project.org>
- Python Package Index
<https://pypi.org/>
- Why not in SAS?

Inspirations

- Comprehensive T_EX Archive Network
<https://ctan.org/pkg/>
- Comprehensive R Archive Network
<https://cran.r-project.org>
- Python Package Index
<https://pypi.org/>
- Why not in SAS?



SAS package¹

A **SAS package** is an automatically generated, single, stand alone zip file containing organised and ordered code structures, created by the developer and extended with additional automatically generated "driving" files (i.e. description, metadata, load, unload, and help files, etc.)

The purpose of a package is to be a simple, and easy to access, code sharing medium, which allows: on the one hand, to separate the code complex dependencies created by the developer from the user experience with the final product and, on the other hand, reduce developers and users unnecessary frustration related to a remote deployment process.

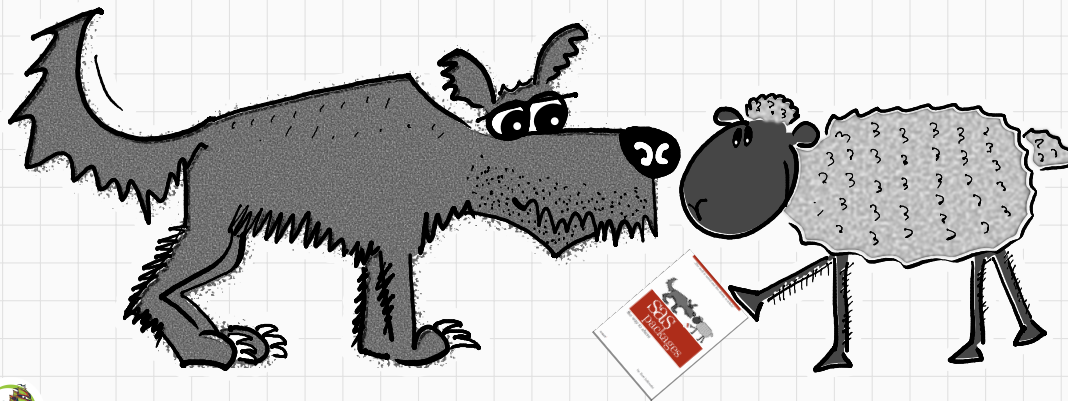
¹ The idea presented here should not be confused with other occurrences of "package" concept which can be found in the SAS ecosystem e.g., Proc DS2 packages, SAS/IML packages, SAS ODS packages, SAS Integration Technologies Publishing Framework packages, or even an Enterprise Guide *.egp file.

SAS Packages Framework

The **SAS Packages Framework** (SPF) is a "pack" of macros, which allows to *use* and to *develop* SAS packages.

- %listPackages(),
 - %installPackage(),
 - %verifyPackage(),
 - %helpPackage(),
 - %previewPackage(),
 - %loadPackage(), %loadPackageS(),
 - %loadPackageAddCnt(),
 - %unloadPackage(),
 - %extendPackagesFileref(),
- and
- %generatePackage().

the User & the Developer



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[®]

„The wolf is sated and the sheep is whole”

SASPAC Archive



sas
packages

the way to share

github.com/SASPAC

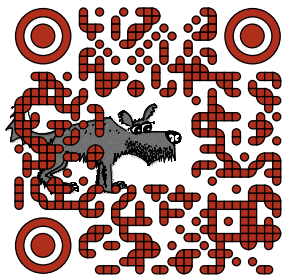
by Bart Jablonski

ORLY⁷

Workshop Materials

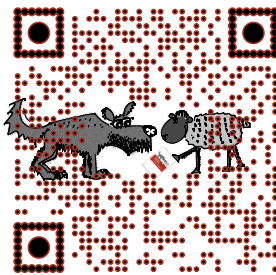


github.com/yabwon/How-SASPackages



sas
packages

the way to share



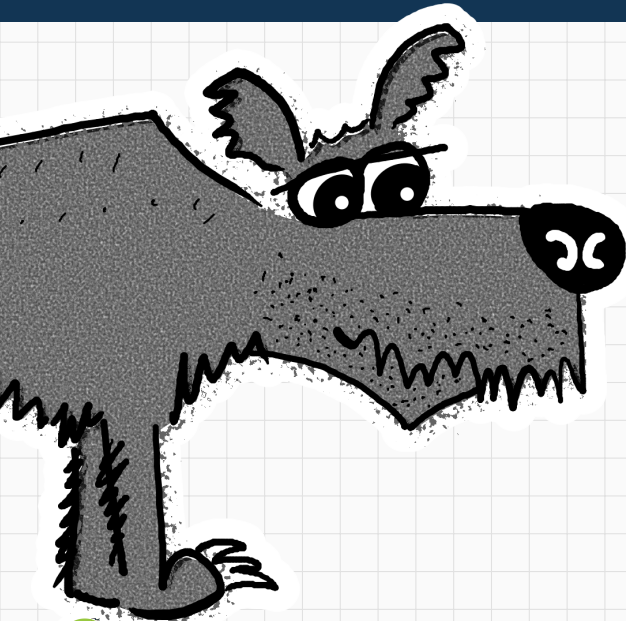
sas
packages

the way to share



sas
packages

the way to share



Files and folders

- Create a folder for packages, e.g. C:/PCKG
- Install the framework: SPFinit.sas*

Code

```
filename packages "C:/PCKG";  
%include packages(SPFinit.sas);  
  
%installPackage(packageName)  
%<help/verify/preview>Package(packageName)  
  
%loadPackage(packageName)
```

What next?

- Read the log
- Use the package ;-)

Files and folders

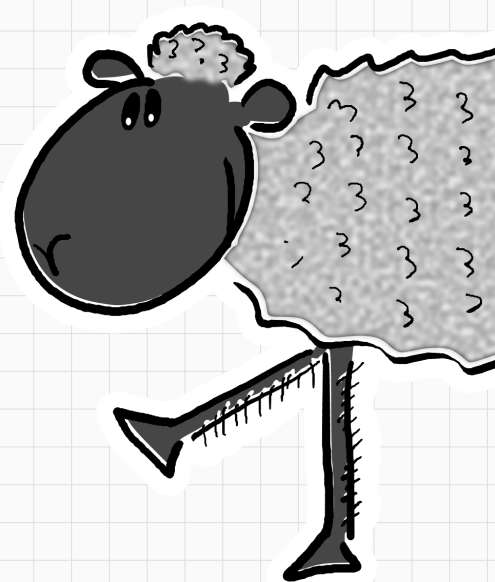
- Install SPFinit.sas* file, e.g. into /pckg
- Create package folder, e.g. /pckg/myPackage
- Prepare package files and the description file.

The Code

```
filename packages "/pckg";  
%include packages(SPFinit.sas);  
%generatePackage(filesLocation=/pckg/myPackage)
```

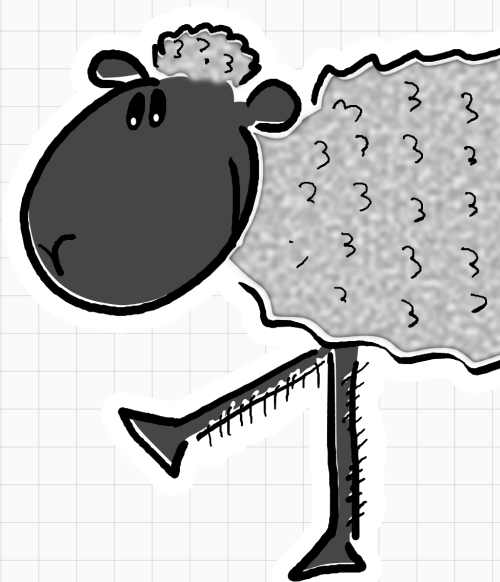
The Result

- Read the summary and the log
- Check tests results
- Share the package ;-)



Available types

libname,
macro,
functions,
formats,
imlmodule,
proto,
data, lazydata,
exec, clean,
casludf,
kmfsnip,
addcnt,
(test).



description.sas

```
Type: Package
Package: ThePackageName
Title: Short description, a single sentence.
Version: x.y.z
Author: Fname1 Lname1 (xxx1@yyy.zz), Fname2 Lname2 (xxx2@yyy.zz)
Maintainer: Fname3 Lname3 (xxx3@yyy.zz)
License: XYZ17
Encoding: UTF8
```

```
Required: "Base SAS Software", "SAS/ACCESS Interface to ABC"
```

```
ReqPackages: "somePackage (3.14)", "otherPackage (42)"
```

DESCRIPTION START:

```
  Lorem ipsum dolor sit amet, ThePackageName consec tetur
  adipis cingelit. Nullamdapibus lacus a elit congue
  elementum. Suspendisse iaculis ipsum nec ante luctus
  volutpat. Donec iaculis laoreet tristique.
```

DESCRIPTION END:



dziękuję

