



METAPROGRAMMING WITH SAS – ADVANTAGES AND CHALLENGES

16 November 2017 Bloemfontein
Tiaan van der Spuy

PAREXEL[®]
YOUR JOURNEY. OUR MISSION.[®]

WHAT IS METAPROGRAMMING?

“A programming technique in which computer programs have the ability to treat programs as their data. It means that a program can be designed to read, generate, analyze or transform other programs, and even modify itself while running.”

Source : [Wikipedia](#)

WHY METAPROGRAMMING?

- To produce SAS TLF programs quicker and more accurate

OUR METHOD



ANATOMY OF A SAS PROGRAM

PART 1

- Initializing information and environment setup

**NO PROGRAMMER
INPUT/TIME NEEDED**

PART 2

- Written by programmers up until final dataset.

**PROGRAMMER
INPUT/TIME**

PART 3

- ODS and PROC REPORT code for tables and listings. ODS, PROC TEMPLATE and SGRENDER code for figures.

**SOME PROGRAMMER
INPUT/TIME NEEDED**

PART 1

```
* =====
* -----;
* Study information;
* -----;
* Client          : MyClient;
* Protocol no     : Happy-Pill-1337;
* PAREXEL code    : PXL111111;
* Target transfer : Dry Run x-Draft x;
* -----;
* Program information;
* -----;
* Program name     : program.sas;
* Current author   : <user>;
* Date and time    : DDMONYYYY HH:MM;
* Description      : Table xx.x.x: xxxxxxxxxxxxxxxxxxxxxxxxxx
* Version         : Dry Run x v0.1;
* -----;
* SAS environment;
* -----;
* SAS version      : 9.4;
* Session encoding : UTF-8;
* =====

* -----;
* Set all paths used;
* -----;
%LET path_prog = %str(.../programs/);
%LET path_log  = %str(.../logs/);
%LET path_doc  = %STR(.../docs/);
* =====

* -----;
* Create log file header and route log/output to file;
* -----;
PROC PRINTTO NEW LOG = "%path_log.program.log";
RUN;
PROC PRINTTO NEW PRINT = "%path_log.program.out";
RUN;
* =====

* -----;
* Set programming environment;
* -----;
%INCLUDE "%path_prog.autoexec.sas" / LRECL=32767;
* =====

* -----;
* Access programmer created formats;
* -----;
%INCLUDE "%path_prog.formats.sas" / LRECL=32767;
* =====
```

PART 2

```
❑ PROC SORT DATA = soure.ADxx OUT=workds1 (WHERE=(xxx = 'Y')) ;  
    BY xxx;  
RUN;  
  
.;  
.;  
.;  
.;  
  
❑ DATA final;  
    SET workds9;  
RUN;
```

PART 2 - DRAWBACKS

- The resultant SAS code is unstructured and needs to be formatted.
- Quoted strings may contain words similar to SAS keywords.
- Only `*;` comments pull through to the resultant SAS code.
- Time consideration – code effectively runs twice

PART 3

```
/* -----ReportIt----- */
_rep_dsin=
/* ===== */
_rep_emptyds= _rep_emptydscenter= _rep_emptydscol=
/* ===== */
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
_rep_sh = _rep_id = _rep_wr = _rep_fm = _rep_lb = _rep_ju =
/* ===== */
_rep_lblgrp= _rep_varsgrp=
_rep_lblgrp= _rep_varsgrp=
_rep_lblgrp= _rep_varsgrp=
/* ===== */
_rep_CARRY=
/* ===== */
_rep_NOSPLIT=
_rep_LINEBREAK=
_rep_PGBREAK=
/* ===== */
_rep_SPANPAGE=
/* ===== */
_rep_ADDROWS=
/* ===== */
_rep_ADDFOOT=
_rep_ADDFOOT=
_rep_ADDFOOT=
_rep_ADDFOOT=
/* ===== */
_rep_ADDTIT= _rep_ADDTITVAR= _rep_ADDTITEND= _rep_ADDTITJUST=
_rep_ADDTIT= _rep_ADDTITVAR= _rep_ADDTITEND= _rep_ADDTITJUST=
_rep_ADDTIT= _rep_ADDTITVAR= _rep_ADDTITEND= _rep_ADDTITJUST=
/* -----ReportItEnd----- */
```

- SAS macro?
- TOC and bookmark code
- Standardized final datasets

PART 3 - DRAWBACKS

- **Extended run time**
- **Debugging can be frustrating**

ADVANTAGES

- **Productivity**
- **Uniformity**
- **Protecting intellectual property**
- **Ensure quality and protecting confidentiality**
- **Automatic update of information and version control in real time**

THANK YOU