

Boosting the efficiency of data review by using SAS Integration Technologies and VBA

Giuseppe Di Monaco

11.OCT.2016



Inspired by **patients.**
Driven by **science.**

How many times have you wanted to have
“a **robust** and **fast** method to deal with
errors caused by incorrect macro input
parameters?”

Agenda

I

- Errors caused by incorrect macro input parameters
- Defect prevention
- Primary objective
- Secondary objective

B

- Data Review tool
- Data Review tool improvement
- Configuration file

P

- Example macro configuration

S

- First Approach (Classical)
- Second Approach (Prerequisite)
- Final Approach (Graphical User Interface + Prerequisite)

S

- Problem I wanted to solve
- Solution
- Demo

INTRODUCTION

:E

- Costs of repairing incorrect variable names
- Necessity of reviewing **lengthy** log files to **identify** issues

D

- A process for:
 - **Detection**
 - **Isolation**
 - **Identification**
 - **Support** for **resolution** of errors

P

- To show how a “**minimum**” combination of:
 - **VBA** (Visual Basic for Applications)
 - **SAS Base**
 - **SAS IOM** (Integrated Object Model)

can be used to create a **module** (**Prerequisite**) that performs **fault-blocking** actions to avoid runtime errors caused by unexpected input data

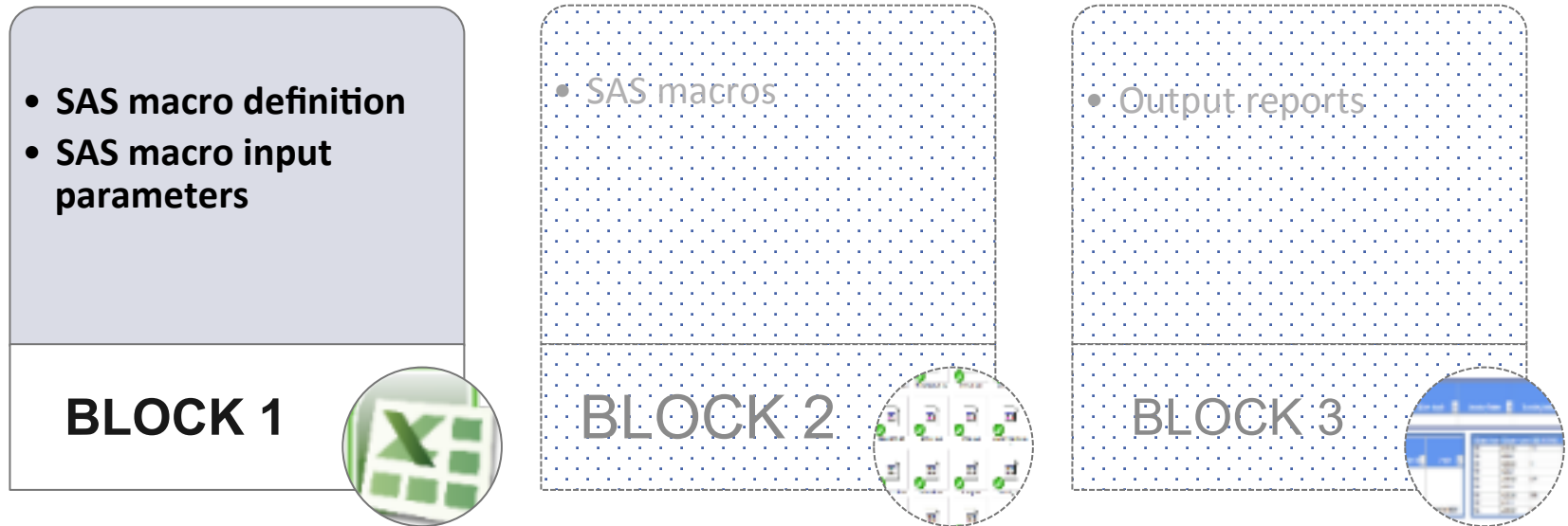
S

- To improve the process and user experience – adding a GUI

BACKGROUND 1/3

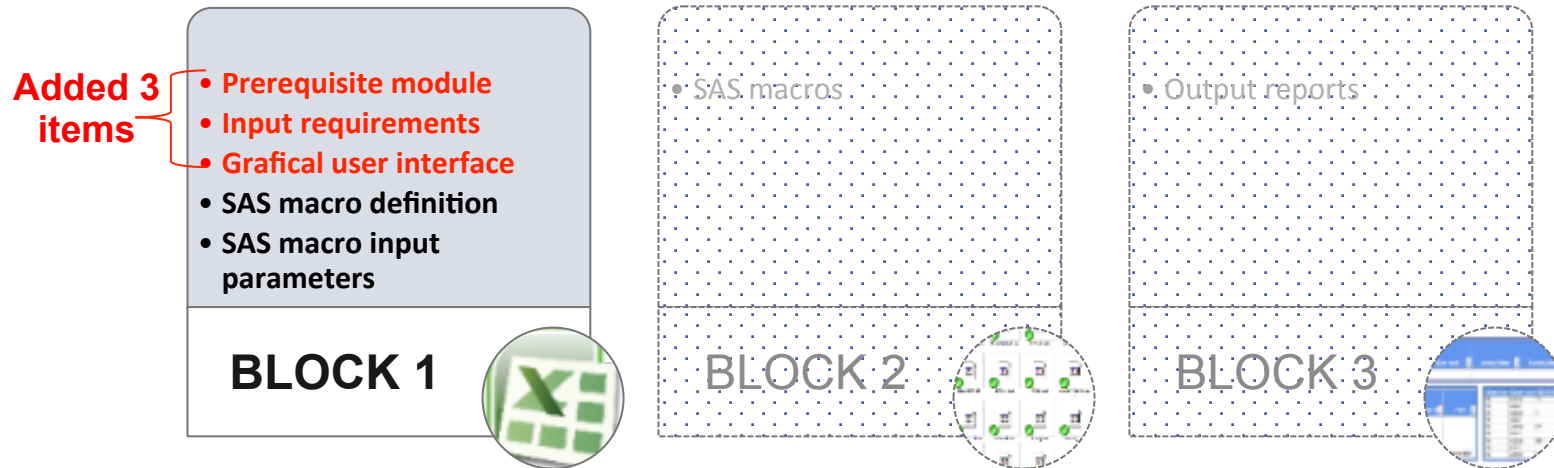
D

- Is an application that ensures that the data is as clean and consistent as possible
- Comprises three main blocks



BACKGROUND 2/3

D



- Prerequisite Module & Input requirements
 - Ensure the **correctness** of the input parameters stored in BLOCK 1
 - Are used by the SAS macros (contained in BLOCK 2)
 - In case of errors, **corrective action** can be taken **before** SAS macros run
- GUI
 - **Reduces** the **steep learning curve** needed to make changes to **configuration file**
 - **Reduces** the risk of making mistakes during manual changes

From now on this presentation will focus on BLOCK 1

BACKGROUND 3/3

C

- For brevity we concentrate on **TRACESINGLEVARIABLE** macro
- Designed to **trace** a value of a variable **across domains**.
 - In this case, it is used to trace the value of **ARMCD** from **SDTM.DM** to all appropriate **ADaM** datasets
- This macro expects:
 - **4 Required** parameters (but only 2 checked by Prerequisite **TARGETDS**; **SECDSET** were checked earlier in the process)
 - **1 Conditional** parameter, meaning that it should be equal to "X" only if all required parameters exist
 - **All others are Optional** (such as **CONDITION** and **ODISPVAR**)

TYPE	SDTM AND ADaM
SELECT	X
MACRO	TRACESINGLEVARIABLE
DESCRIPTION	Check traceability of ARMCD between SDTM dataset DM and all ADaM datasets
TARGETDS	AD. ALL
TARGETV	ARMCD
SECDSET	SD.DM
SECVAR	ARMCD
WHERETRG	
CONDITION	
WHERESEC	
LIST	
ODISPVAR	
GRADE	
.....	
Parameter n	

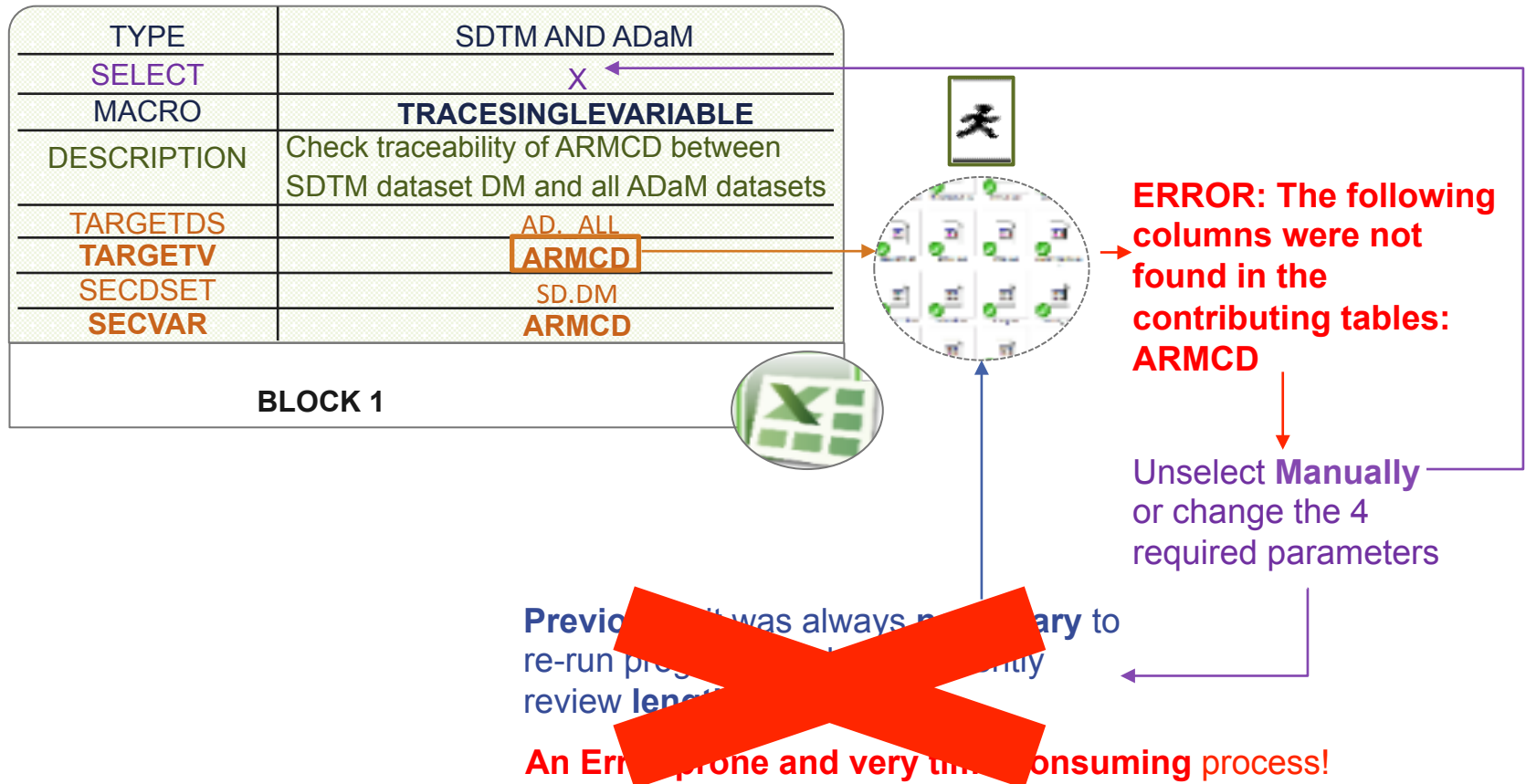
BLOCK 1



PROBLEM – INCORRECT INPUT

E

- A very simple scenario where an error can occur while running a SAS program
- ARMCD variable not found in any of the datasets present in AD library. (TARGETDS = AD._ALL_)



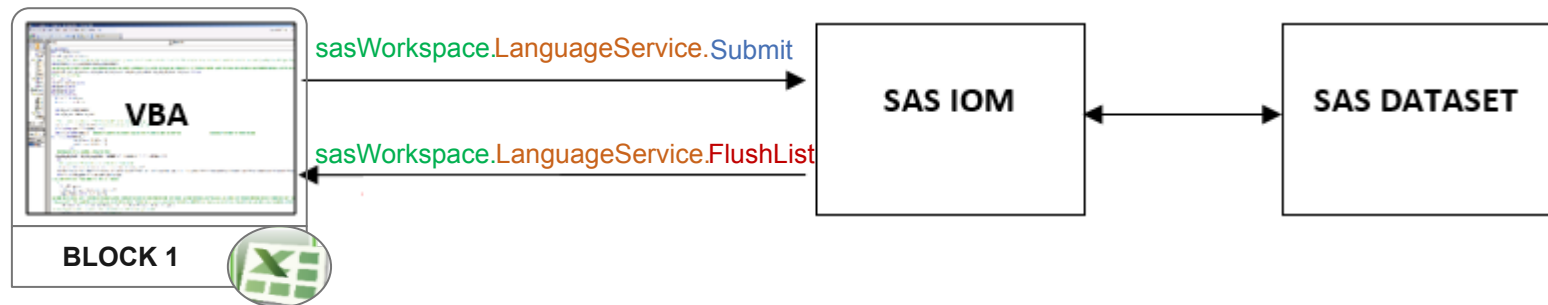
SOLUTION 1/5

)F

- A macro should not run at all
 - Many SAS programmers use SAS code aimed at **stopping** macro execution
 - ERROR: *****
 - ERROR: The TARGET variable (ARMCD) is not in the AD._ALL_ dataset(s) you requested
 - ERROR: Please check the spelling of the variable and/or dataset name
 - ERROR: *****
 - This is **not efficient** when running the application over dozens of macros, each with multiple parameters

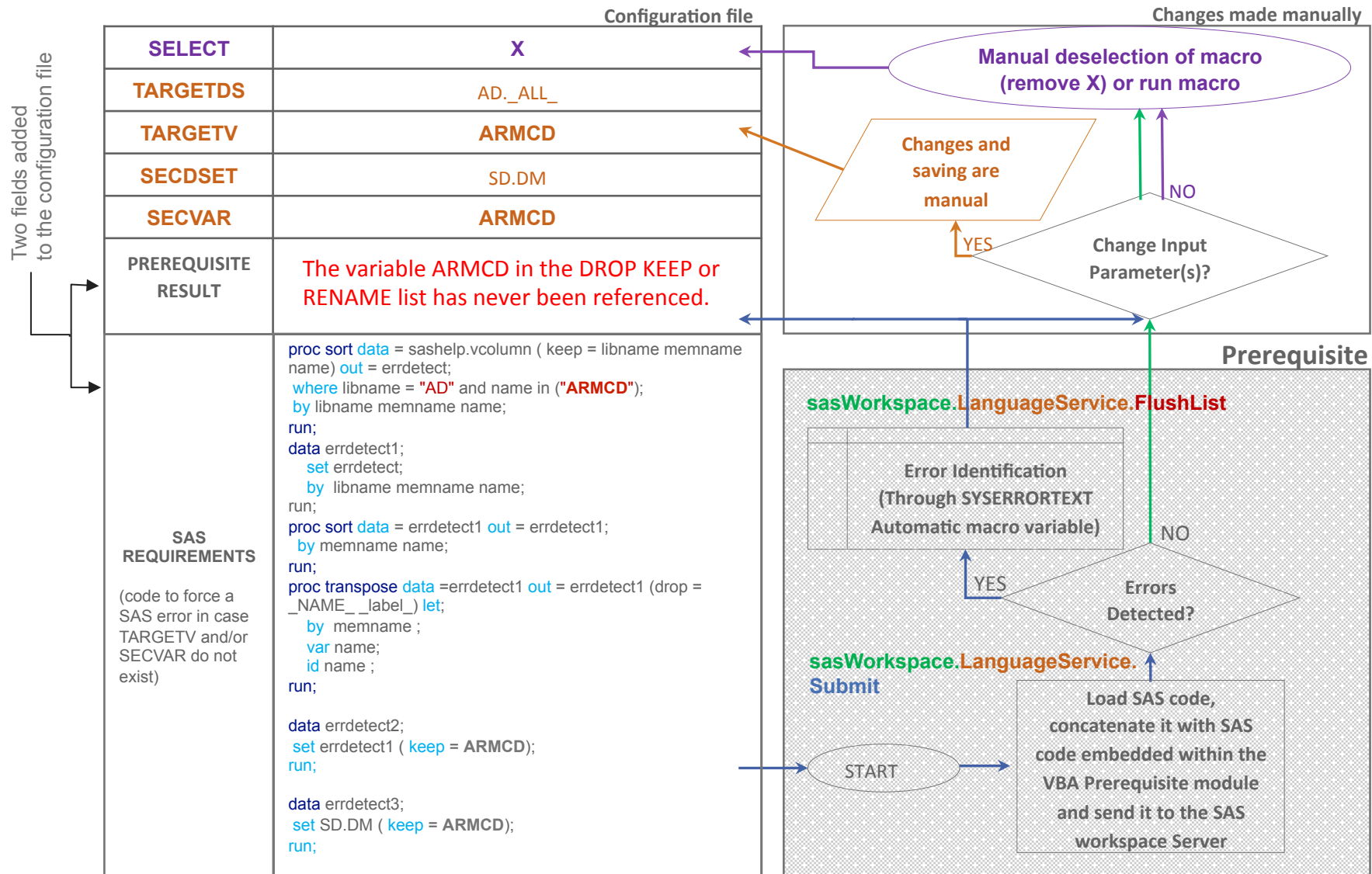
)S

- Prerequisite (run and retrieve information from SAS)
 - Was developed using VBA and the SAS IOM



- **SAS workspace**. A session with the SAS System (similar to the execution of base SAS software in batch mode)
- **LanguageService** Object. A service for submitting SAS language
- **Submit**: A method for submitting SAS language statements provided by the client
- **FlushList**: A method for retrieving list lines that were spooled by program execution

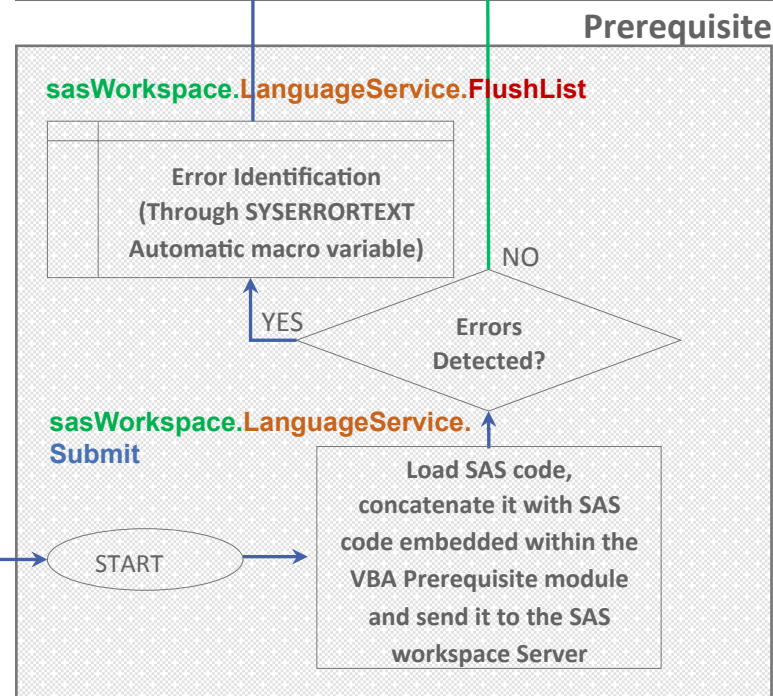
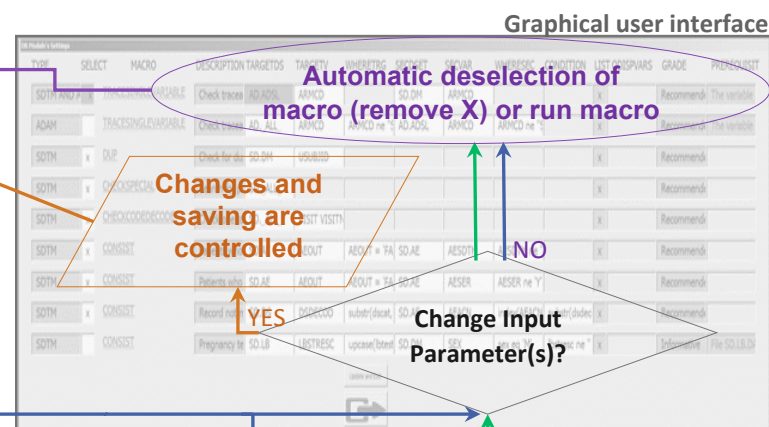
)S



)F

Two fields added
to the configuration file

Configuration file	
SELECT	X
TARGETDS	AD._ALL_
TARGETV	ARMCD
SECDSET	SD.DM
SECVAR	ARMCD
PREREQUISITE RESULT	The variable ARMCD in the DROP KEEP or RENAME list has never been referenced.
SAS REQUIREMENTS	<pre>proc sort data = sashelp.vcolumn (keep = libname memname name) out = errdetect1; where libname = "AD" and name in ("ARMCD"); by libname memname name; run; data errdetect1; set errdetect1; by libname memname name; run; proc sort data = errdetect1 out = errdetect1; by memname name; run; proc transpose data =errdetect1 out = errdetect1 (drop = _NAME_ _label_) let; by memname ; var name; id name ; run; data errdetect2; set errdetect1 (keep = ARMCD); run; data errdetect3; set SD.DM (keep = ARMCD); run;</pre> (code to force a SAS error in case TARGETV and/or SECVAR do not exist)



SOLUTION 4/5

)F

- GUI increases usability of the Data Review Tool

- 2 Isolation**
- 6 Macro Label** has a direct link to the SharePoint web page
- 3 Dark Grey** non-missing and non-enterable fields.
- 5 White** enterable fields
- 4 Light Grey** missing and non-enterable fields
- 1 Error Identification**

NOTE: if you receive ADSL which contains ARMCD. You can either keep TARGETDS=AD._ALL_ or change it to AD.ADSL and reselect with an "X"

F

- Update macro parameters (The interface enables changes to the configuration and returns the result to the spreadsheet)

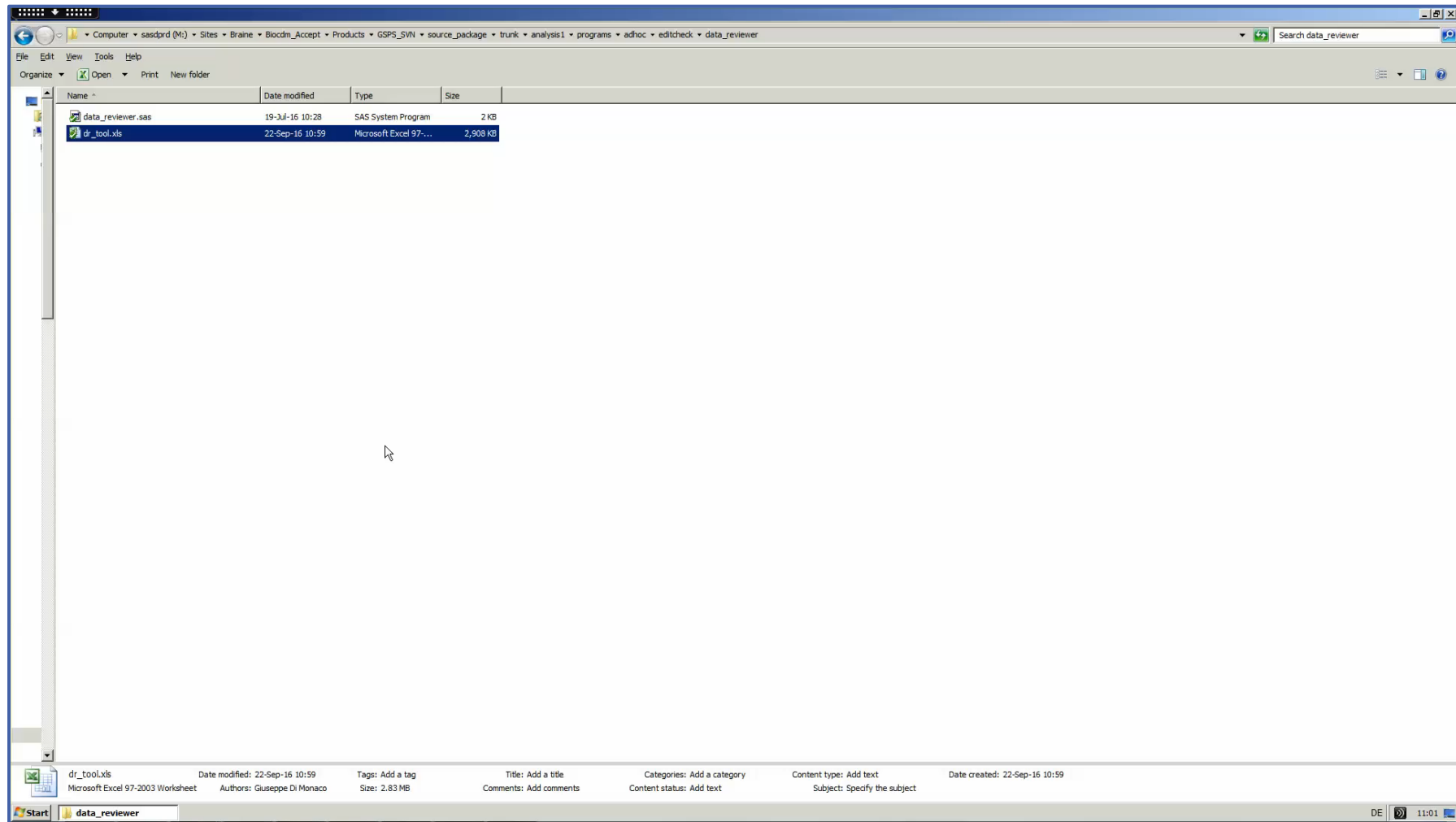
TYPE	SDTM AND ADaM
SELECT	X
MACRO	TRACESINGLEVARIABLE
DESCRIPTION	Check traceability of ARMCD between SDTM dataset DM and all ADaM datasets
TARGETDS	AD.ADSL
TARGETTV	ARMCD
SECDSSET	SD.DM
SECVAR	ARMCD

BLOCK 1

SUMMARY 1/2

14

D



SUMMARY 2/2

P

- Avoid the following error-prone and time-consuming process
 1. **Manually** reviewing logs for errors caused by **faulty input** parameters
 2. **Manual** update of input parameters
 3. **Waiting** for the application to finish running (**again**)
 4. **Restart** from point 1

S

- **Checking** and **updating** of SAS macro parameters **before** running SAS macros
- A combination of **VBA**, **SAS Base**, **SAS IOM** and a **configuration file** aimed at:
 - Detecting
 - Isolating
 - Identifying
 - Supporting **Error resolution**
- **Note:** This method
 - is not **limited** to checking the input parameters of SAS macros
 - could be applied in any situation where there is a need to check the inputs expected by programs

Questions?

Thanks!