



entimo[®]

Application Development

AD 08

Automated Test Framework for SAS Macros

Sven Prasse

2019-11-08

Agenda

- Introduction
- Test Environment
- Test Case
- Log Analysis
- Result Analysis
- Outlook
- Questions

Introduction

- Agile software development and DevOps are designed to bring software into production as soon as possible
- In nightly (or faster) builds new versions are deployed
- Without any automated testing, it is not possible to guarantee a high level of software quality
- The presentation shows how a SAS framework supports the development of SAS Standard macros based on automation of
 - Test execution,
 - Collection and archiving of test results,
 - Comparison of test results against reference results and
 - Provision of summaries and error reports.

Agenda

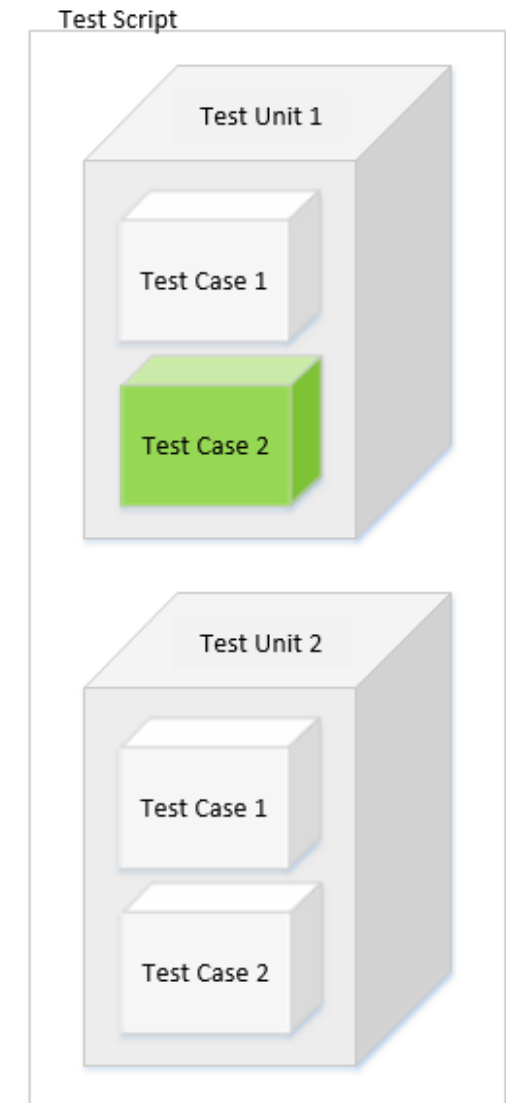
- Introduction
- **Test Environment**
- Test Case
- Log Analysis
- Result Analysis
- Outlook
- Questions

Test Environment

- Configuration Metadata
- Loaded catalogue with the compiled test macros as
 - SAS program controller
 - Log parser
 - Output comparison modules
 - Result storage handler
- Test Objective
 - A loaded program to test
- Test Data
- Test References
 - Program's outputs

Test Script

- Logical segmentation of a test script into test units
- A test unit contains the test cases
- Test unit name and test case name identify a test case

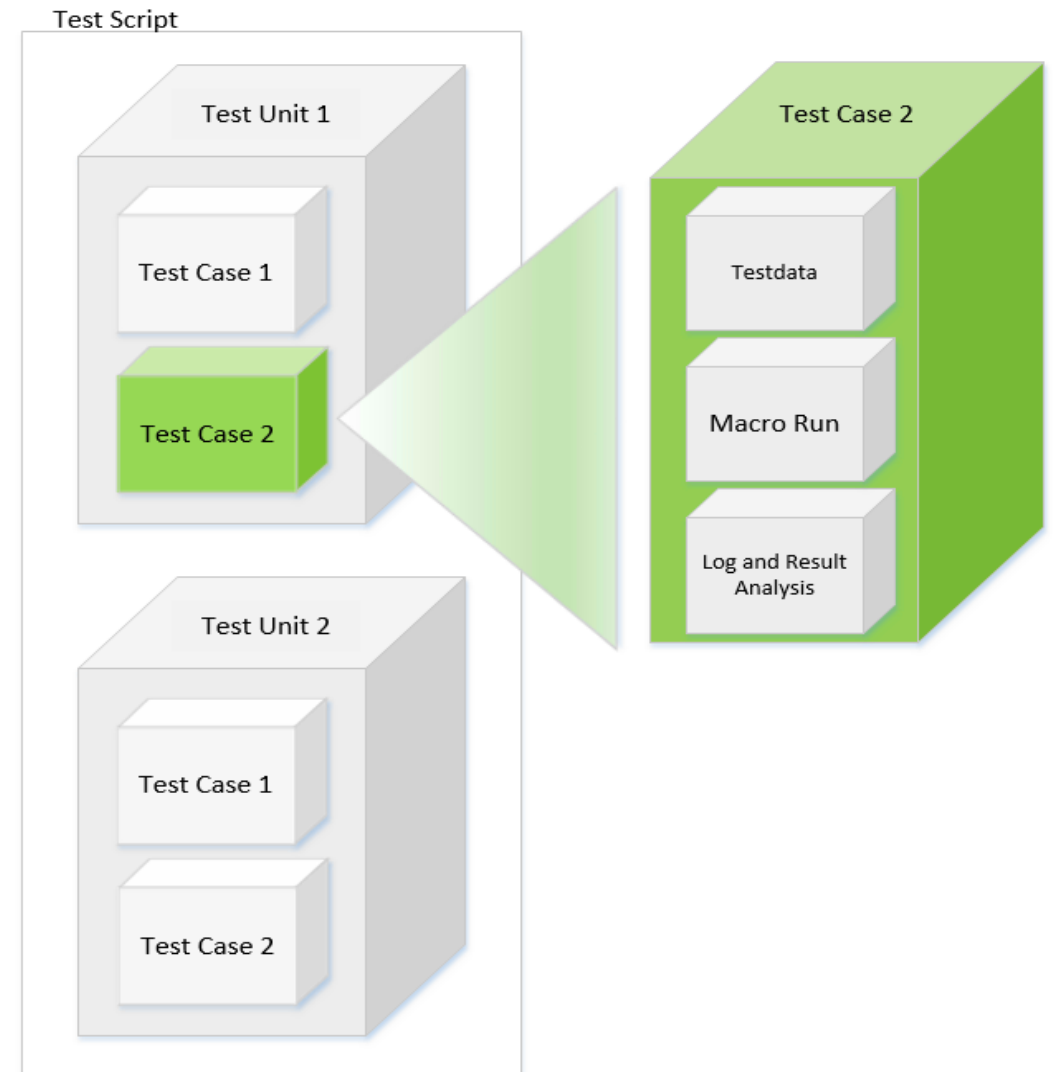


Agenda

- Introduction
- Test Environment
- Test Case
- Log Analysis
- Result Analysis
- Outlook
- Questions

Test Case Processing

- Preparation of the test data
- Run of the macro
 - Store the log
 - Save the outputs
- Log analysis
- Result comparison



Test Case Code Example

- **%NEWTESTCASE**('##A','Missing or partial start dates ');
- **%TESTDATA**();
- **OPTIONS SOURCE**;
- **DATA** inae_testcase_1;
- **SET** TestData.ae_tnc;
- **WHERE** USUBJID in ('TEST_0001_0000000001','TEST_0001_0000000002');
- **IF** substr(AESTDTC, 1, 10) = "2002-06-24" AND AESEQ = 2 **THEN DO**;
- AESTDTC = "";
- **END**;
- **RUN**;
- **OPTIONS NOSOURCE**;
- **%MACCALL**;
- **%LET** MACROCALL = MACRO2TEST (INAE = inae_testcase_1, OutAdamAE = outref.ADAE_TESTCASE_1);
- **%EXPECTRESULT**;
- **%PUT** A clear log is expected;
- **%TESTRESULT**;
- **%COMPARE_SAVE_REFERENCE**(baselib = LibRef, complib = outref, new = ADAEPL);
- **%ENDTESTCASE**(check4UnexpectedLogMsg=Y);

Agenda

- Introduction
- Test Environment
- Test Case
- **Log Analysis**
- Result Analysis
- Outlook
- Questions

Log Analysis

- The expectation of a correct macro run is a “clear” SAS log
- The metadata define global strings to identify an unclear log (e.g., “ERROR:”)
- In addition test case specific messages has to be identified

Definition of Expected Messages

- %PUT statements with brackets {} in the test script define the expected strings
- This avoids extra storage management

%EXPECTRESULT;

%PUT A table of the subjects per study with missing or partial dates is shown;

%PUT {WARNING - Missing or partial start dates in the following AEs};

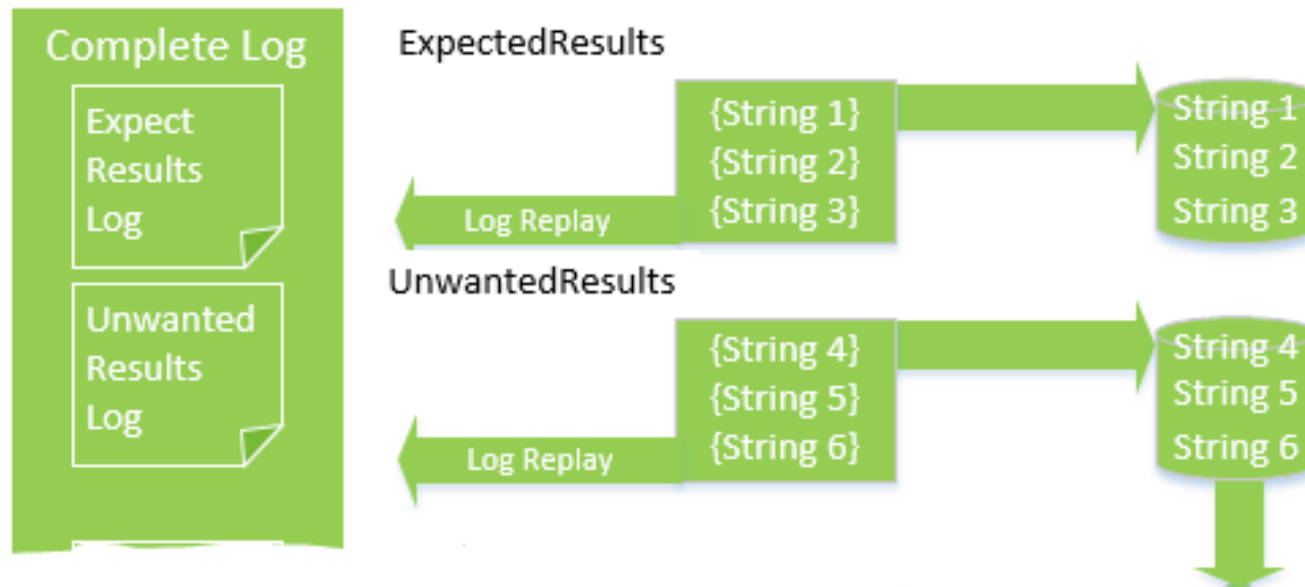
%PUT {*** STUDYID USUBJID AESEQ};

%PUT {-----};

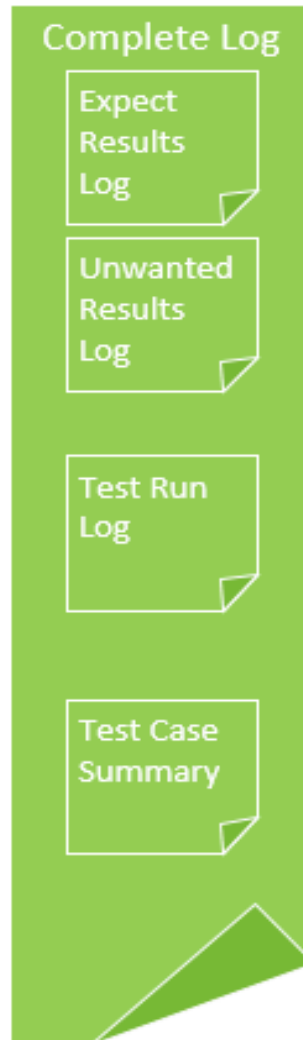
%PUT {*** TEST-0002 TEST_0002_0000001001 1};

Log Analysis

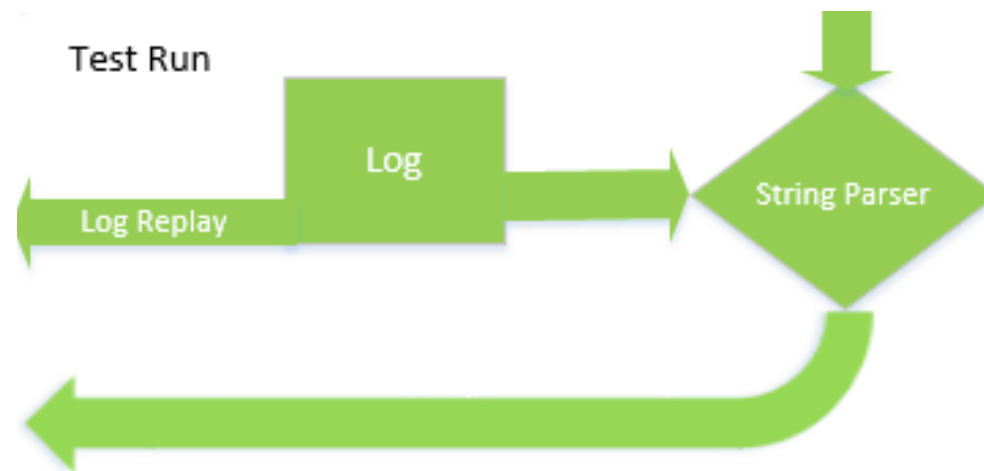
- The redirected log will be parsed for expected and unwanted strings in the corresponding sections



Log of the Macro Run



- The log of the run and the comparison results log will be concatenated to the complete log



Log Comparison

■ Definition

Expected result:

A table of the subjects ...

{WARNING - Missing or partial start dates in the following AEs:};

{*** STUDYID USUBJID AESEQ};

{-----};

{*** TEST-0002 TEST_0002_0000001001 1};

Test result:

* YOU ARE RUNNING THE FOLLOWING MACRO *

* MACRO : MACRO2TEST VERSION = 2.3 *

WARNING - Missing or partial start dates in the following AEs:

*** STUDYID USUBJID AESEQ

*** TEST-0002 TEST_0002_0000001001 1

■ Run

Agenda

- Introduction
- Test Environment
- Test Case
- Log Analysis
- **Result Analysis**
- Outlook
- Questions

Result Comparison

- The output comparison modules are specialized for one specific output type
 - Datasets
 - Text files (e.g. LST Outputs)
 - Graphics
- Dataset comparison is done by a PROC Compare between the result of the macro run and a reference dataset

Result Comparison: Text Files

- In LST outputs may contain information that depends on the run time (e.g. the current date)
- Such an piece of information cannot be stored in a static reference

Respiratory, thoracic and mediastinal disorders	0	2	0	0	0	0	0
Epistaxis	0	2	0	0	0	0	0

FOOT7
FOOT8
Containing data from study
27SEP2016

- E.g.: the footer of an output table contains
 - A fixed string "Containing the data"
 - And the current date

Define Masks

-

Respiratory, thoracic and mediastinal disorders	0	2	0	0	0	0	0
Exhaustion	0	0	0	0	0	0	0

FOOT7
FOOT8

Containing data from stud

27SEP2016

Result Comparison: Graphics (.png)

- For the png format, identical figures have different binary representations on different machines. This depends on the physical devices and the drivers that are used.
- For the figures using SAS GTL, the smartest solution catches
 - the input dataset
 - the source code for PROC SGRENDER
 - and the GTL Template
- The dataset and the two text files can be compared with the references

Agenda

- Introduction
- Test Environment
- Test Case
- Log Analysis
- Result Analysis
- Outlook
- Questions

Outlook

- Automated testing is optimal for agile software development
- With every developing round of the test objective, the test cases cover more and more functionality
- That leads to the fact: quality grows with the features
- After some sprints (software development cycles) the invested time of the development of test cases will return in form of quicker cycles and higher quality of the software
- In addition the automated tests can be used for regression analysis

Thank you for your attention!

Questions and
Answers

