Paper TT04

Another Year, Another Migration: Planning the Move from Entimo entimICE to SAS LSAF

Pantaleo Nacci, AstraZeneca, Barcelona, Spain

ABSTRACT

To better cope with the increasing needs and demands of the evolving pharma scenario, AstraZeneca (AZ) has launched Redefining Clinical Data Flow (RCDF), an ambitious project to completely revamp how clinical research is implemented in the company. As part of this larger project, it has been decided to change our current Statistical Computing Environment (SCE) from entimICE-AZ to SAS® Life Sciences Analytical Framework (LSAF).

This change will require the gradual migration of all clinical study data, programs, deliverables, etc. from the existing platform to a completely different new one, posing a series of issues with maintaining the normal flow of activities.

INTRODUCTION

This paper offers a high-level discussion of the migration, problems envisioned, and then describes some of the more technical aspects and solutions linked to this planned migration, with special focus on possible ways to automate the adaptation of existing SAS programs to the new environment.

Several relevant topics (training, business processes, system support) have already been covered in a paper [2] presented at the 2022 EU PHUSE Connect in Belfast, so I will not touch on them again here.

RCDF IN A NUTSHELL

The RCDF programme aims to identify systems and processes to allow the company to scale up clinical research activities as needed..

THE VISION

The poster below, presented during the virtual 2021 EU PHUSE Connect, describes the overall goal of the project:

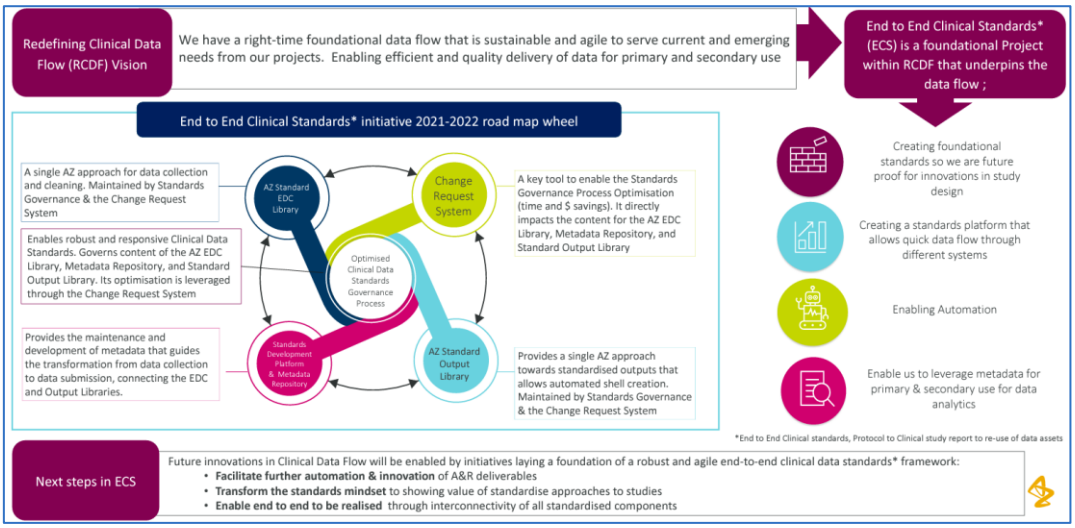


Fig. 1 The RCDF overall vision

Since then, more details have been hammered out on the actual software solutions, their integrations and related processes to be used, but the overall picture remains the same as initially envisaged.

RCDF IMPACT ON SYSTEMS

As hinted earlier the changes in both systems and processes, planned and already implemented, are many and cover a wide variety of topics: this paper will concentrate on those relevant to the roles of Clinical Programmer, Statistical Programmer and Statistician.

A&R SYSYEMS INVOLVED

Currently most activities by both Data Management (DM) and Analysis & Reporting (A&R) are performed in entimICE-AZ, the AZ-specific implementation of the entimICE® DARE environment, developed by Entimo® AG described in [2].

As a result of the changes, all A&R activities and the DM ones requiring access to SAS will move from entimICE-AZ to LSAF over a prolonged period, during which both systems will be actively used. This means that at least part of the existing user base will need to be trained on LSAF and the new business processes while still working in entimICE-AZ.

INTEGRATION WITH OTHER SYSTEMS

Within the new scenario LSAF is only one piece of a very complex puzzle, and several new integrations are being worked on to allow the correct and unimpeded flow of data, from data collection to the publishing of results.
A partial list of these other systems, several of them new to AZ, includes Medidata® Rave EDC, several Saama® modules, SAS Viya and a couple of solutions developed internally (Patient Data Landing Zone, Patient Data Retention).
As mentioned earlier the two A&R systems will be used contemporaneously, so, e.g., raw data will need to be directed to either one, then exchanged between them constantly to work on integrated analyses including data from multiple studies.

COMPARISON OF SOURCE AND TARGET A&R SYSTEMS

Both entimICE-AZ and LSAF are modern SCEs, i.e., closed, customisable environments with controlled access and full audit trail, offering an integrated editor to write and execute programs as well as a host of other tools and capabilities. Having been designed at different times though, they have different technical approaches to their architectures, and that is reflected in how some common issues have been addressed.
In AZ both are accessed via a Citrix desktop with restricted Internet access, so that downloaded files remain in a controlled environment and cannot be easily shared outside of it.

MAIN HIGH-LEVEL DIFFERENCES

One architectural difference especially relevant to programmers is the type of interaction with the SAS System: while entimICE-AZ acts as a frontend, launching external sessions when needed, LSAF is built on top and fully integrated with it. This allows an easier interaction with the programming environment while developing code; so the datasets and views in the SASHELP library are directly accessible in LSAF, but not in entimICE-AZ. On top of that the LSAF program editor is very similar to the one offered in SAS Studio, while the one in entimICE-AZ is the standard one provided in Eclipse (other external editors can be used, but then the system integration is lost).
While in entimICE-AZ the location of all objects needed to correctly execute a program must be specified via ‘links’, no such need exists for LSAF: this is a definite advantage from the programmers’ point of view, especially when accessing data for multiple studies placed in different locations.
The default encoding of SAS sessions differs too, being (W)LATIN1 for entimICE-AZ and UTF-8 for LSAF: this has possible implications when migrating datasets between systems, and steps must be taken to make sure no data corruption happens. Alternate SAS session encodings can be manually selected in entimICE-AZ, but not in LSAF.
Another (very welcome) limitation in LSAF compared to entimICE-AZ is that it is impossible to use the ‘X’ command and similar ones in SAS programs: this ensures that the audit trail is always engaged and can record any action performed by users.
Finally, entimICE-AZ is physically located in a Swedish AZ data centre and maintained internally, while LSAF is cloud-based and managed by SAS Institute.

FOLDER STRUCTURES

The entimICE-AZ folder structure was designed almost 10 years ago and made the most of the characteristics offered by the system. The A&R top branch (‘cdar’, for Clinical Data Analysis & Reporting, shared by both DM and A&R staff), is organized according to a project/study/functional area/reporting event/delivery/execution area hierarchy, and forces a high level of standardization by reusing the same basic structures in multiple places. In practical terms, when creating a new standard folder structure the user must always do it via a node-specific context menu entry, with the ability to add new sub-folders and naming them at will only in a very restricted number of cases. This characteristic allows the maintenance of a very ordered structure, without the variations and multiplication of sub-folders usually linked to the personal styles of the users working in a certain study area.
Moving to LSAF, some details of the new folder structure, organized instead according to a functional area/project/study/reporting event/execution area/delivery hierarchy, are still being defined to address a few remaining source locations, like restricted deliveries and exploratory areas. Here too the ability of normal users to create new folders has been restricted to the lower levels of the structure, while only IT staff can create new contexts (high-level folders) in the repository. Hopefully these limitations will keep the folder structure variability under control.
While the new standard structure is similar to the existing one, some of the changes introduced, like the swapping of delivery and execution area levels, the removal of the ‘dev’ execution area and the different location of doc folders, complicate the migration of objects from entimICE-AZ to LSAF.

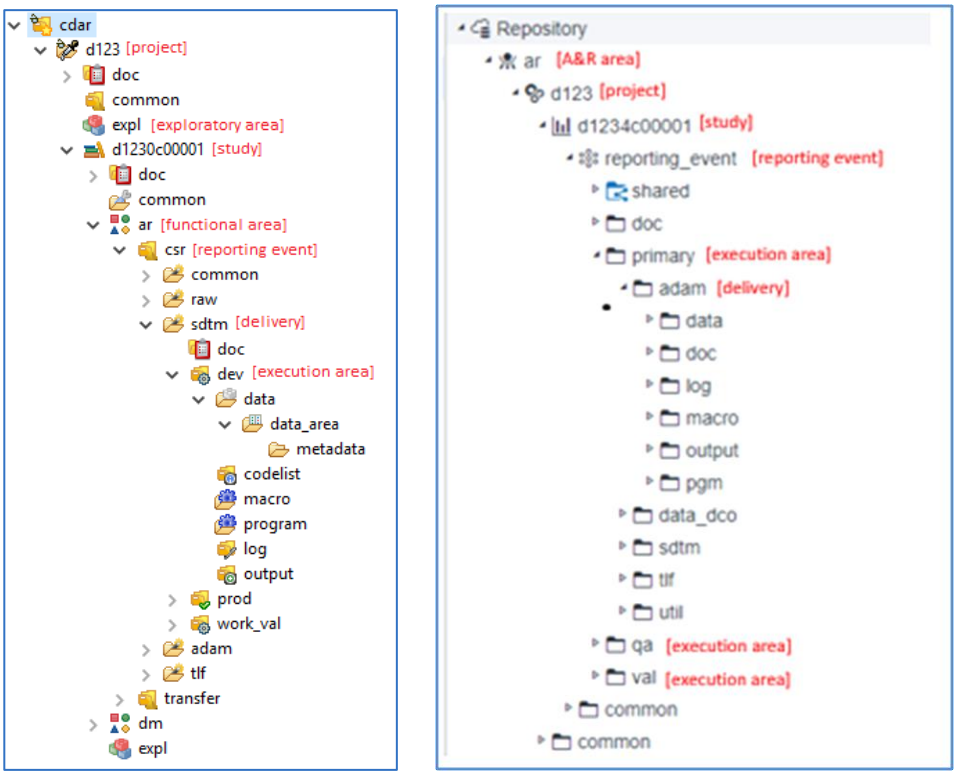


Fig. 2 The entimICE-AZ (left) and LSAF (right) standard folder structures

ENVIRONMENT VARIABLES

Both systems provide the users with several automatic environment variables, surfaced as macro variables to SAS programs. When properly used these variables allow users to write highly portable code, since they avoid to a great extent the need to hardcode paths or other details, like the username of the programmer who produced a certain output.

In entimICE-AZ several of these variable and their values are listed at the beginning of log files, together with other information like the host information, the version of SAS and the session encoding. As can be seen below the number of these automatic macro variables is quite consistent, even if only a subset is of use for most programmers:

```
1 --- SCSM --- SCSM 2.3  on SAS 9.04.01M7P08062020/LIN X64 Linux/latin1 sesklentssdv01 for icesas
2 vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv (No. 2)  09OCT2024 12:34:48.154
3 166       +options notes;
4 □11                                                    The SAS System
5
6 167       +%let _exec_userid = kbvt788;
7 168       +%let _exec_userlastname = Nacci;
8 169       +%let _exec_userfirstname = Pantaleo;
9 170       +%let _exec_programpath = root/cdar_/d123/d1230c00001/ar/csr/sdtm/dev/program;
10 171      +%let _exec_programname = test.sas;
11 172      +%let _exec_programversion = 1.0;
12 173      +%let _exec_programstatus = development;
13 174      +%let _exec_programfileversion = -1;
14 175      +%let _exec_programtags = ;
15 176      +%let _exec_programlastuser = kbvt788;
16 177      +%let _exec_programlastaction = Object has been created;
17 178      +%let _exec_programlastmodified = 2024-10-09T09:20:36;
18 179      +%let _exec_namesprefix = _entimICEsys_;
19 180      +%let _exec_snippet_run = N;
20 181      +%let _exec_log_area = root/cdar_/d123/d1230c00001/ar/csr/sdtm/dev/log;
21 182      +%let _exec_output_area = root/cdar_/d123/d1230c00001/ar/csr/sdtm/dev/output;
22 183      +%let _exec_run_background_info = client|direct_call|normal_controller_first_call;
23 184      +options fmtsearch=(WORK LIBRARY);
24 185      +options nomstored;
25 186      +%let study_name = d1230c00001;
26 187      +%let mp_compound_name = ;
27 188      +%let functional_area_name = ar;
28 189      +%let execution_area = dev;
29 190      +%let delivery_name = sdtm;
30 191      +%let reporting_event_name = csr;
31 192      +%let ms_study_phase = ;
32 193      +%let project_name = d123;
33 194      +%let corp_name = cdar_pn;
```

Fig. 3 Automatic macro variables and system information in entimICE-AZ

In LSAF the number of automatic environment variables is more limited, but the logs show the same system information (apart of course from the session encoding, which is fixed to UTF-8):

```
NOTE: Copyright (c) 2016 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software 9.4 (TS1M7 MBCS3170)
      Licensed to ASTRAZENECA UK LTD LSAF SAS CLOUD PRE-PROD, Site [redacted].
NOTE: This session is executing on the Linux 4.18.0-305.76.1.el8_4.x86_64 (LIN X64) platform.
```

```
4          %let _SDDUSR_=%bquote(kbvt788);
5          %let _SASWS_ = %nrstr("/lustrefs/lsafshared/SASWorkspaces/kbvt788");
6          %let _SASWS_ = %qsubstr(&_SASWS_, 2, %length(&_SASWS_) - 2);
7          %let _SASUSRWS_ = %nrstr("/lustrefs/lsafshared/SASWorkspaces/kbvt788/Users/kbvt788");
8          %let _SASUSRWS_ = %qsubstr(&_SASUSRWS_, 2, %length(&_SASUSRWS_) - 2);
9          %let _SASINSTANCE_ = %nrstr("azklsafpreprod.ondemand.sas.com");
10         %let _SASINSTANCE_ = %qsubstr(&_SASINSTANCE_, 2, %length(&_SASINSTANCE_) - 2);
11         options nosource;

NOTE: DATA statement used (Total process time):
      real time           1.01 seconds
      cpu time            0.00 seconds


18

/*************************************************************************************************************
 * Submission Start: test.sas
 * Oct 09, 2024 09:57:30 GMT by kbvt788
 *************************************************************************************************************
2                                                    The SAS System                    Wednesday,

19         options nosyntaxcheck errorcheck=strict notes;
20         ;*';*";*/;*); run; quit;
21
22         %let _SASFILELOCATION_ = %nrstr("/ar_training/d123/d1234c00001/dr1/primary/sdtm/pgm");
23         %let _SASFILELOCATION_ = %qsubstr(&_SASFILELOCATION_, 2, %length(&_SASFILELOCATION_) - 2);
24         %let _SASFILEPATH_ = %nrstr("/ar_training/d123/d1234c00001/dr1/primary/sdtm/pgm/test.sas");
25         %let _SASFILEPATH_ = %qsubstr(&_SASFILEPATH_, 2, %length(&_SASFILEPATH_) - 2);
```

Fig. 4 Automatic macro variables and system information in LSAF

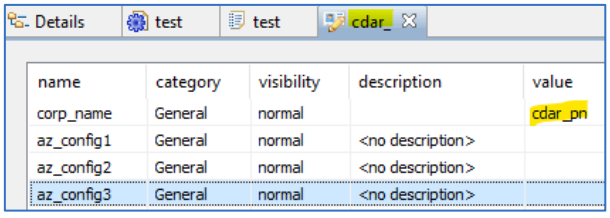
THE %SETUP MACRO

An environment initialisation macro, named simply setup.sas, has been developed in entimICE-AZ: heavily based on the entimICE automatic variables mentioned earlier, its aim is to create a programming environment as standardised as possible, with a set of predefined common library names, additional macro variables and other assets which programmers can use as a basis when writing programs.

```
GLOBAL CORP_NAMEX cdar_
GLOBAL DELIVERY_NAMEX sdtm
GLOBAL EXECUTION_AREAX dev
GLOBAL EXTV lst
GLOBAL FUNCTIONAL_AREA_NAMEX ar
GLOBAL GFONT tahoma
GLOBAL ODS_OUT lst1
GLOBAL PROJECT_NAMEX d123
GLOBAL REPORTING_EVENT_NAMEX csr
GLOBAL STUDYDIR root/cdar_/d123/d1230c00001/ar/csr
GLOBAL STUDY_NAMEX d1230c00001
GLOBAL SYS_ROOT_PATH root
```

Fig. 5 Macro variables defined by setup.sas in entimICE-AZ

In particular, parallel versions of several macro variables are created, ending with an ‘X’, derived from the physical path rather than from each node’s properties.



name	category	visibility	description	value
corp_name	General	normal		cdar_pn
az_config1	General	normal	<no description>	
az_config2	General	normal	<no description>	
az_config3	General	normal	<no description>	

Fig. 6 Physical names vs. node properties in entimICE-AZ

The %setup macro has been ported to LSAF with several changes, mostly linked to the differences in folder structure and system terminology, e.g., ‘NAME’ has been often replaced with ‘CONTEXT’. A handful of the entimICE automatic macro variables are also recreated by the macro.

```
GLOBAL DELIVERY_UNITX sdtm
GLOBAL EXECUTION_AREAX primary
GLOBAL EXTV
GLOBAL FUNCTION_CONTEXTX ar_training
GLOBAL ODS_OUT
GLOBAL PROJECT_CONTEXTX d123
GLOBAL REPORTING_EVENT_CONTEXTX dr1
GLOBAL STUDYDIR /lustrefs/lsafshared/SASWorkspaces/kbvt788/ar_training/d123/d1234c00001/dr1
GLOBAL STUDY_CONTEXTX d1234c00001
GLOBAL SYS_ROOT_PATH ▯▯lustrefs▯lsafshared▯SASWorkspaces▯kbvt788▯
GLOBAL TITLEPATH
GLOBAL WHRTITL
GLOBAL _EXEC_OUTPUT_AREA ▯▯lustrefs▯lsafshared▯SASWorkspaces▯kbvt788▯/ar_training/d123/d1234c00001/dr1/primary/sdtm/output
GLOBAL _EXEC_PROGRAMNAME test.sas
GLOBAL _EXEC_PROGRAMPATH ▯▯lustrefs▯lsafshared▯SASWorkspaces▯kbvt788▯▯▯ar_training▯d123▯d1234c00001▯dr1▯primary▯sdtm▯pgm▯
GLOBAL _EXEC_USERID kbvt788
```

Fig. 7 Macro variables defined by setup.sas in LSAF

To increase flexibility, on both platforms setup.sas can automatically include and execute another macro (localsetup.sas), which can thus be used to further tailor the programming environment.

Details | localsetup

```
%macro localsetup;

options sasautos=("&studydir/&delivery_namex/prod/macro"
                  "&studydir/&delivery_namex/&execution_areax/macro"
                  "&studydir/common/prod/macro"
                  "&studydir/common/&execution_areax/macro"
                  "root/&corp_namex/&project_namex/&study_namex/common/code_library/prod/macro"
                  "root/&corp_namex/&project_namex/&study_namex/common/code_library/&execution_areax/macro"
                  "root/&corp_namex/&project_namex/common/code_library/prod/macro"
                  "root/&corp_namex/&project_namex/common/code_library/&execution_areax/macro"
                  "root/cdar/common/gmd/prod/macro"
                  "root/cdar/common/gmd/&execution_areax/macro"
                  "root/cdar/common/ecb/prod/macro"
                  "root/cdar/common/ecb/&execution_areax/macro"
                 sasautos);

%mend;
```

Fig. 8 Example of localsetup.sas in entimICE-AZ

IDENTIFYING AND APPLYING CHANGES IN SAS PROGRAMS

As described above, in the LSAF version of %setup.sas the names of several macro variables have changed, and one macro variable has disappeared completely.

When we designed the process to adapt automatically entimICE-AZ SAS programs to the new system, the decision was taken to apply as few changes to the source code as possible, so that when comparing the two versions it would be easier to spot differences.

RECREATE MISSING VARIABLES

The first step was to develop a special version of localsetup.sas, recreating some of the entimICE-AZ macro variables not already defined in LSAF:

```
%macro localsetup;

%* Recreate entimICE-AZ macro variables to ease porting of SAS programs from there to LSAF ;
%* NB The positions of &delivery_namex and &execution_areax need to be swapped in the paths of entimICE-AZ programs ;

%global corp_namex project_namex study_namex functional_area_namex reporting_event_namex delivery_namex;

%let corp_namex            = &function_contextx;
%let project_namex         = &project_contextx;
%let study_namex           = &study_contextx;
%let functional_area_namex = ;
%let reporting_event_namex = &reporting_event_contextx;
%let delivery_namex        = &delivery_unitx;
%*let execution_areax       = &execution_areax;

options insert=sasautos=("&studydir/primary/&delivery_namex/macro"
                         "&studydir/&execution_areax/&delivery_namex/macro"
                         "&studydir/primary/util/macro"
                         "&studydir/&execution_areax/util/macro"
                         "&sys_root_path/&corp_namex/&project_namex/&study_namex/common/macro"
                         "&sys_root_path/global_tools/corp/common/current/primary/macro"
                         "&sys_root_path/global_tools/corp/logcheck/current/primary/macro"
                         );
%mend;
```

Fig. 9 Version of localsetup.sas to support entimICE-AZ SAS programs migration to LSAF

IDENTIFY MODIFICATIONS

Once all basic entimICE-AZ macro variables were available, we concentrated on a small set of programs to identify the most common situations where code changes were needed. The resulting list is a work in progress, as the inspection of more programs from other studies will most probably highlight further common code, which can then be automatically modified.

These are some of the changes identified until now:

a. Whenever values for project, study, reporting event and delivery are hardcoded, replace them with the corresponding macro variables
b. Since programs have been developed on a variety of operating systems using different representations for end-of-line, make it homogeneous
c. Remove dots used to terminate macro variable names when not needed
d. Remove unnecessary blanks
e. Make sure the existing calls to %setup execute localsetup.sas (the call needs to be added manually when not already present)
f. Exchange hardcoded entimICE-AZ folders (root, cdar, ar) with macro variable
g. Change execution area values (dev, prod, work_???) into either macro variables or corresponding LSAF values
h. Swap position of delivery and execution_area
i. Change 'common' into 'util', but only at delivery level
j. Change 'program' into 'pgm'
k. Change default location of SAS formats catalogs
l. Change default location of input CDASH datasets
m. Relocate 'doc' folders
n. Change normal (SBCS) string functions into their K-version (MBCS) counterparts

As I mentioned earlier, this list is a work in progress, and the expectation is that it will expand and change as more programs are examined in preparation for the migration.

APPLY CHANGES

There are several possibilities to implement the identified changes, using, e.g., either an advanced text editor with scripting capabilities or a specialised tool.

In our case we developed a Windows batch script using the standard tool gsar (General Search And Replace) to apply them.

The figure below shows the first part of the batch file:

```
rem The following commands search & replace text in entimICE-AZ SAS programs to help with their adaptation to AZ LSAF
rem The utility gsar.exe (https://gnuwin32.sourceforge.net/packages/gsar.htm) is used to perform the actual substitutions
rem
rem Fix any project/study/reporting event/delivery-specific hardcodings
gsar -s/d000/ -r/:x26project_namex/ -i -o *.sas
gsar -s/d0000c00017/ -r/:x26study_namex/ -i -o *.sas
gsar -s/dr1/ -r/:x26reporting_event_namex/ -i -o *.sas
gsar -s/d567/ -r/:x26project_namex/ -i -o *.sas
gsar -s/d5671c00006/ -r/:x26study_namex/ -i -o *.sas
gsar -s/expl_synthetic/ -r/:x26reporting_event_namex/ -i -o *.sas
gsar -s/d798/ -r/:x26project_namex/ -i -o *.sas
gsar -s/d7980c00003/ -r/:x26study_namex/ -i -o *.sas
gsar -s/jreview/ -r/:x26delivery_namex/ -i -o *.sas
rem DO NOT MODIFY BELOW THIS LINE
rem Change all EOL to *nix
gsar -du -i -o *.sas
rem Remove unnecessary dots at the end of standard macro variable names
gsar -s:x26sys_root_path./         -r:x26sys_root_path/ -i -o *.sas
gsar -s:x26corp_namex./            -r:x26corp_namex/ -i -o *.sas
gsar -s:x26project_namex./         -r:x26project_namex/ -i -o *.sas
gsar -s:x26study_namex./           -r:x26study_namex/ -i -o *.sas
gsar -s:x26functional_area_namex./ -r:x26functional_area_namex/ -i -o *.sas
gsar -s:x26reporting_event_namex./ -r:x26reporting_event_namex/ -i -o *.sas
gsar -s:x26delivery_namex./        -r:x26delivery_namex/ -i -o *.sas
gsar -s:x26execution_areax./       -r:x26execution_areax/ -i -o *.sas
gsar -s:x26studydir./              -r:x26studydir/ -i -o *.sas
rem Remove unnecessary trailing spaces
gsar -s:x25setup:x20 -r:x25setup -i -o *.sas
gsar -s):x20;        -r); -i -o *.sas
rem Add option to automatically execute localsetup macro in setup macro call
gsar -s:x25setup;              -r:x25setup(exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup();            -r:x25setup(exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup:x0a           -r:x25setup(exec_localsetup=Y);:x0a -i -o *.sas
gsar -s:x25setup(version=251); -r:x25setup(version=251,exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup(version=252); -r:x25setup(version=252,exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup(version=253); -r:x25setup(version=253,exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup(version=260); -r:x25setup(version=260,exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup(version=300); -r:x25setup(version=300,exec_localsetup=Y); -i -o *.sas
gsar -s:x25setup(version=301); -r:x25setup(version=301,exec_localsetup=Y); -i -o *.sas
rem Substitute root with _SASWS_, but only when specifying path of localsetup.sas
gsar -slocalsetup=root/ -rlocalsetup=:x26_sasws_/ -i -o *.sas
rem Substitute root with &sys_root_path anywhere else
gsar -sroot/            -r:x26sys_root_path/ -i -o *.sas
```

Fig. 10 Implementation of identified changes using the gsar tool

MANUAL INSPECTION

The resulting migrated versions of the programs still need to be thoroughly inspected, even when they seem to run without issues. Hopefully only minor changes are still needed to make the program run correctly in the new environment, but this depends on the level of standardisation adopted when writing the original program.

Any command trying to open a shell to the underlying operating system needs to be analysed and substituted with one of the many functions available in SAS.

CONCLUSION

As I stated at the end of my 2022 paper, the migration of clinical data from one A&R system to another over a prolonged period is always a complex affair, requiring a deep technical and operational understanding of both the source and target systems to be executed successfully.

In this occasion some aspects of the target system are still being discussed, which makes it difficult, e.g., to finalise the details of where each object in the source system will be placed after the migration.

Technical and architectural differences between the two systems added a further layer of difficulty, as there was only partial direct knowledge of the target one within the company.

Fig. 11 Basic points to keep into consideration when planning a data migration

REFERENCES

[1] Grabowska M, Harris B. Redefining Clinical Data Flow at AstraZeneca. PHUSE EU Connect 2021, Virtual

[2] Nacci P. Migrating a large users base from multiple legacy systems to entimICE-AZ: issues and lessons learned. PHUSE EU Connect 2022, Belfast

[3] Miller T, Nacci P, Schepers A, Woo W. The Clinical Data Repository Initiative at Novartis Vaccines and Diagnostics: a Revolution. FDA/PhUSE Computational Science Symposium (CSS) 2012, Silver Spring, MA

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Pantaleo Nacci
AstraZeneca
Av. Diagonal, 615
08028 Barcelona
Spain
Email: pantaleo.nacci@astrazeneca.com