



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

Pantaleo Nacci, Global Clinical Solutions,
BioPharmaceuticals Clinical Operations,
R&D, AstraZeneca, Barcelona, Spain



Agenda

1. Introduction
2. Redefining Clinical Data Flow (RCDF) in a Nutshell
3. RCDF Impact on Systems
4. Comparison of Source and Target A&R Systems
5. Identifying and Applying Changes in SAS Programs
6. Conclusions



1

Introduction



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
- Special focus is 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 presented during the 2022 EU PHUSE Connect in Belfast



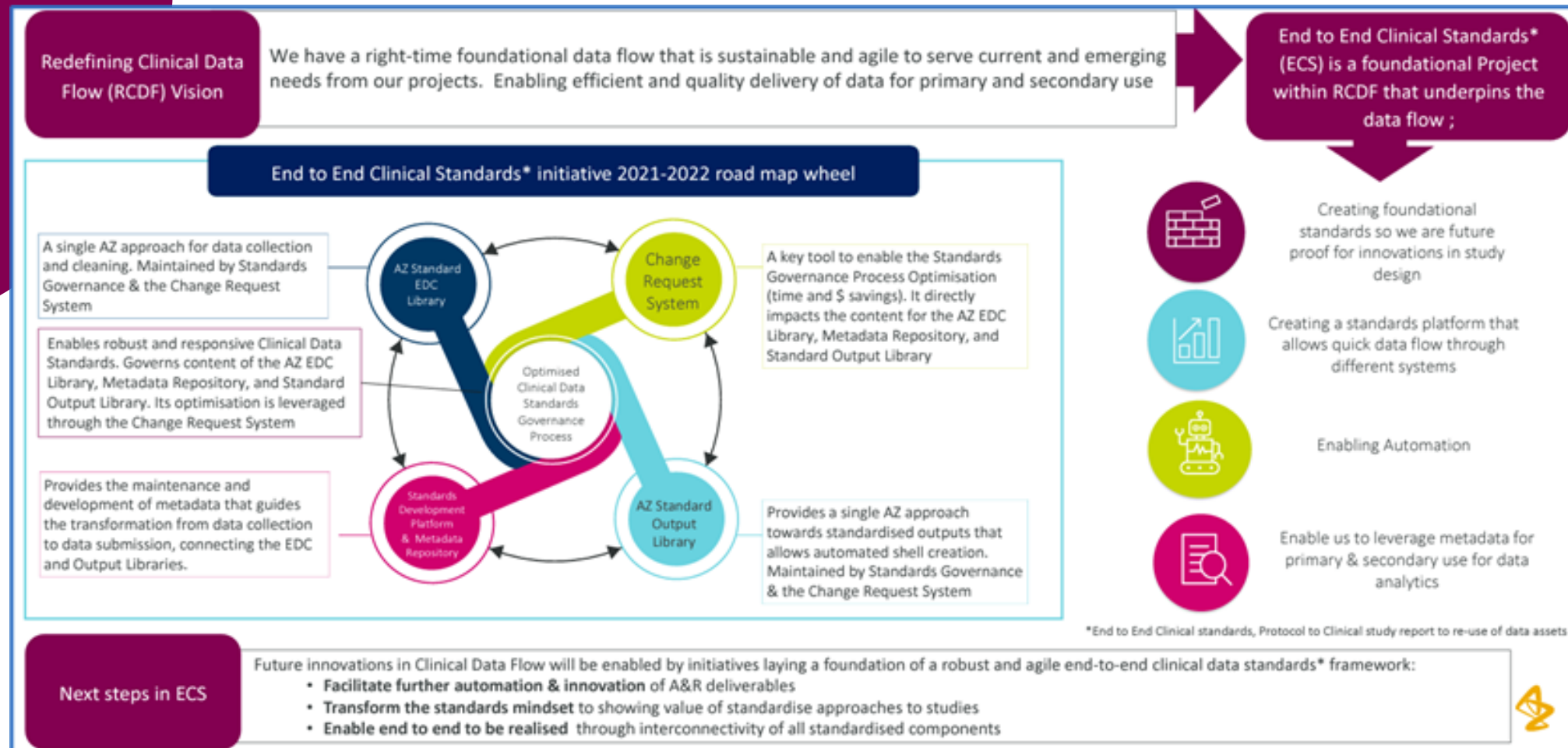
2

RCDF in a Nutshell



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 poster below, presented during the virtual 2021 EU PHUSE Connect, describes the RCDF vision.
- Since then, more details have been hammered out, but the overall picture remains the same.



3

RCDF Impact on Systems



RCDF Impact on Systems

- Changes in systems and processes, both planned and already implemented, are many and cover a wide variety of topics
- This presentation concentrates on those relevant to the roles of Clinical Programmer, Statistical Programmer, and Statistician
- 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, by Entimo® AG
- All A&R activities and the DM ones requiring access to SAS will move from entimICE-AZ to LSAF over a prolonged period
 - Part of the existing user base will need to be trained on LSAF and the new business processes while still working in entimICE-AZ
- LSAF is only one piece of a very complex puzzle, and several new integrations are being worked on
 - E.g., raw data will need to be directed to one of the two systems and then exchanged between them constantly to work on integrated analyses



4

Comparison of Source and Target A&R Systems



Comparison of Source and Target A&R Systems

- Both entimICE-AZ and LSAF are modern Statistical Computing Environments (SCE): closed, customisable environments with controlled access and full audit trail, offering a host of other tools and capabilities
- Having been designed at different times, they have different technical approaches to their architectures
 - This is reflected in how some common issues have been addressed
- In AstraZeneca both systems 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, allowing an easier interaction with the programming environment while developing code
 - E.g., the datasets and views in the SASHELP library are accessible interactively in LSAF, but not in entimICE-AZ
 - The LSAF program editor is similar to the one offered in SAS Studio, while in entimICE-AZ it is the standard one provided in Eclipse
- In entimICE-AZ the location of all objects needed by a program must be specified via 'links', while no such need exists for LSAF
 - In my experience this is one of the hardest concepts for users of entimICE to comprehend



Main High-level Differences (cont)

- The default encoding of SAS sessions is (W)LATIN1 for entimICE-AZ and UTF-8 for LSAF
 - This has implications when migrating datasets between systems or receiving them from external vendors
- In LSAF it is impossible to use the 'X' command and similar ones to open a shell to the underlying operating system in SAS programs
 - This ensures that the audit trail is always active and can correctly record any action performed by users
- entimICE-AZ is physically located in a Swedish AZ data centre and maintained internally, while LSAF is cloud-based and managed by SAS Institute

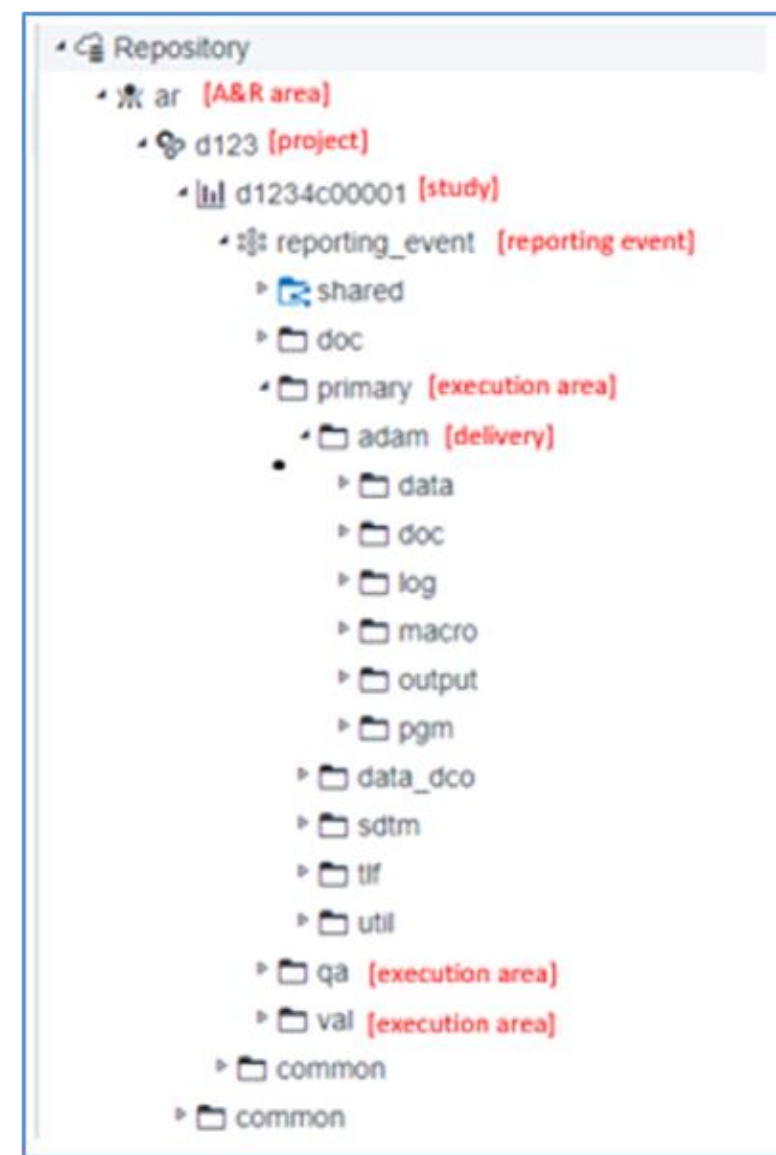
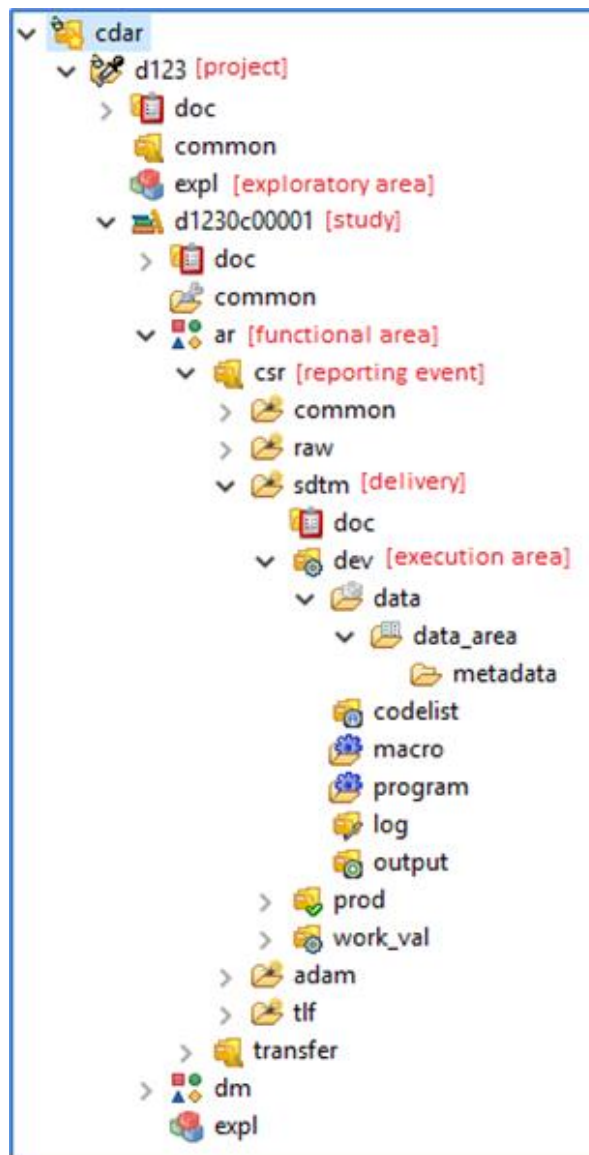


Folder Structures

- The entimICE-AZ folder structure was designed almost 10 years ago to make the most of the characteristics offered by the system
 - The A&R top branch ('cdar', Clinical Data Analysis & Reporting) is organized according to a [project]/[study]/[functional area]/[reporting event]/[delivery]/[execution area] hierarchy
 - It forces a high level of standardization by reusing the same basic structures in multiple places
 - New folders can only be created via context menu entries, and their naming is fixed in several cases
- The new LSAF folder structure is still being completely defined
 - The A&R top branch ('ar') is organized according to a [project]/[study]/[reporting event]/[execution area]/[delivery] hierarchy
 - Only IT staff can create new contexts (high-level folders) in the repository
 - Users are able to create new folders only at the lower levels of the repository



Folder Structures (cont)



Left: entimICE-AZ folder structure; right: LSAF folder structure



Folder Structures (cont)

- While the new LSAF standard structure is broadly similar to the entimICE-AZ one, some of the changes introduced complicate the migration of objects from entimICE-AZ to LSAF, e.g.
 - the swapping of [delivery] and [execution area] levels
 - the removal of the 'dev' execution area
 - the different location of 'doc' folders



Environment Variables – entimICE-AZ

- Both systems provide several automatic environment variables, available as macro variables in SAS programs
- These variables allow users to write highly portable code
- In entimICE-AZ several of these variable and their values are listed at the beginning of log files
 - The logs show other useful information like the operating system, the version of SAS and the session encoding

```

1 --- SCSM --- SCSM 2.3 on SAS 9.04.01M7P08062020/LIN X64 Linux/latin1 sesklientssdv01 for icessa
2 vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv (No. 2) 09OCT2024 12:34:48.154
3 166 +options notes;
4 Q11 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;
```

Environment Variables – LSAF

- In LSAF the number of automatic environment variables is more limited
- The logs show the same system information (apart 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
```

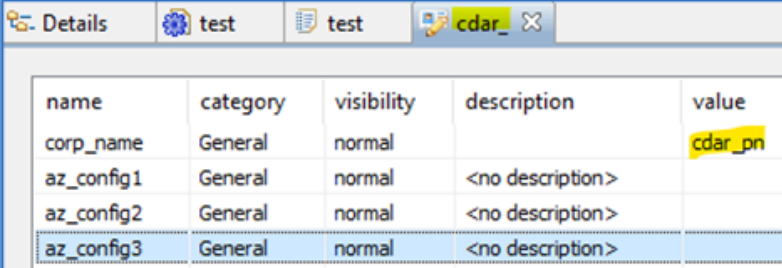
```
18  
  
/*****  
 * Submission Start: test.sas  
 * Oct 09, 2024 09:57:30 GMT by kbvt788  
 *****/  
  
2                                     The SAS System                               Wednesday,  
  
19     options nosyntaxcheck errorcheck=strict notes;  
20     ;*';*";*/;*) run; quit;  
21  
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);
```



The %Setup Macro

- An environment initialisation macro, named 'setup.sas', has been developed in entimICE-AZ
- 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 to be used in programs
- Parallel versions of several macro variables are created, ending with an 'X', derived from the physical path rather than from each node's properties

```
GLOBAL CORP_NAME% cdar_  
GLOBAL DELIVERY_NAME% sdtm  
GLOBAL EXECUTION_AREA% dev  
GLOBAL EXTV lst  
GLOBAL FUNCTIONAL_AREA_NAME% ar  
GLOBAL GFONT tahoma  
GLOBAL ODS_OUT lst1  
GLOBAL PROJECT_NAME% d123  
GLOBAL REPORTING_EVENT_NAME% csr  
GLOBAL STUDYDIR root/cdar_/d123/d1230c00001/ar/csr  
GLOBAL STUDY_NAME% d1230c00001  
GLOBAL SYS_ROOT_PATH root
```



The screenshot shows a SAS configuration window with a tab labeled 'cdar_'. It contains a table with the following data:

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>	



The %Setup Macro (cont)

- The 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

```
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
```



The %Setup Macro (cont)

- 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
- The example below is from entimICE-AZ



```
1 @macro localsetup;
2
3 options sasautos=("&studydir/&delivery_name/prod/macro"
4                  "&studydir/&delivery_name/&execution_areax/macro"
5                  "&studydir/common/prod/macro"
6                  "&studydir/common/&execution_areax/macro"
7                  "root/&corp_name/&project_name/&study_name/common/code_library/prod/macro"
8                  "root/&corp_name/&project_name/&study_name/common/code_library/&execution_areax/macro"
9                  "root/&corp_name/&project_name/common/code_library/prod/macro"
10                 "root/&corp_name/&project_name/common/code_library/&execution_areax/macro"
11                 "root/cdar/common/gmd/prod/macro"
12                 "root/cdar/common/gmd/&execution_areax/macro"
13                 "root/cdar/common/ecb/prod/macro"
14                 "root/cdar/common/ecb/&execution_areax/macro"
15                 sasautos);
16
17 @mend;
```



5

Identifying and Applying Changes in SAS Programs



Identifying and Applying Changes in SAS Programs

- As described earlier, in the LSAF version of setup.sas the names of several macro variables have changed, and one macro variable has disappeared completely
- When designing the process to adapt automatically SAS programs written in entimICE-AZ to work in the new system, we decided to apply as few changes to the source code as possible
 - The aim is to make it easier to spot meaningful differences when comparing the two versions



Recreate Missing Variables

- The first step was to recreate some of the missing entimICE-AZ macro variables taking advantage of the localsetup.sas functionality mentioned earlier
- A specially tailored LSAF version of that macro was developed

```
1 %macro localsetup;
2
3  /* Recreate entimICE-AZ macro variables to ease porting of SAS programs from there to LSAF ;
4  /* NB The positions of &delivery_namex and &execution_areax need to be swapped in the paths of entimICE-AZ programs ;
5
6  %global corp_namex project_namex study_namex functional_area_namex reporting_event_namex delivery_namex;
7
8  %let corp_namex          = &function_contextx;
9  %let project_namex      = &project_contextx;
10 %let study_namex         = &study_contextx;
11 %let functional_area_namex = ;
12 %let reporting_event_namex = &reporting_event_contextx;
13 %let delivery_namex      = &delivery_unitx;
14 /*let execution_areax    = &execution_areax;
15
16 options insert=sasautos=("&studydir/primary/&delivery_namex/macro"
17                          "&studydir/&execution_areax/&delivery_namex/macro"
18                          "&studydir/primary/util/macro"
19                          "&studydir/&execution_areax/util/macro"
20                          "&sys_root_path/&corp_namex/&project_namex/&study_namex/common/macro"
21                          "&sys_root_path/global_tools/corp/common/current/primary/macro"
22                          "&sys_root_path/global_tools/corp/logcheck/current/primary/macro"
23                          );
24 %mend;
```



Identify Modifications

- We run an initial analysis 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, and the inspection of additional programs is expected to highlight further common code to include in the list
- Some of the changes identified until now:
 - Whenever values for [project], [study], [reporting event] and [delivery] are hardcoded, replace them with the corresponding entimICE-AZ macro variables
 - Since programs have been developed on a variety of operating systems using different representations for end-of-line, make it homogeneous
 - Remove dots appended to macro variable names when not needed
 - Remove unnecessary blanks



Identify Modifications (cont)

- Make sure the existing calls to %setup execute localsetup.sas (added manually when not already present)
- Exchange hardcoded entimICE-AZ folders (root, cdar, ar) with macro variables
- Change execution area values (dev, prod, work_???) into either macro variables or corresponding LSAF values
- Swap position of [delivery] and [execution area] in paths
- Change 'common' into 'util', but only at delivery level
- Change 'program' into 'pgm'
- Change default location of SAS formats catalogs
- Change default location of input RAW/CDASH datasets
- Relocate 'doc' folders
- Change normal (Single-Byte Character Set, SBCS) string functions into their K-version (Multi-Byte Character Set, MBCS) counterparts



Apply Changes

- There are several possibilities to implement the 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 snapshot on the following slide shows the first section of the batch file



Apply Changes (cont)

```
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_name/ -i -o *.sas
gsar -s/d0000c00017/ -r/:x26study_name/ -i -o *.sas
gsar -s/drl/ -r/:x26reporting_event_name/ -i -o *.sas
gsar -s/d567/ -r/:x26project_name/ -i -o *.sas
gsar -s/d5671c00006/ -r/:x26study_name/ -i -o *.sas
gsar -s/expl_synthetic/ -r/:x26reporting_event_name/ -i -o *.sas
gsar -s/d798/ -r/:x26project_name/ -i -o *.sas
gsar -s/d7980c00003/ -r/:x26study_name/ -i -o *.sas
gsar -s/jreview/ -r/:x26delivery_name/ -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_name./ -r:x26corp_name/ -i -o *.sas
gsar -s:x26project_name./ -r:x26project_name/ -i -o *.sas
gsar -s:x26study_name./ -r:x26study_name/ -i -o *.sas
gsar -s:x26functional_area_name./ -r:x26functional_area_name/ -i -o *.sas
gsar -s:x26reporting_event_name./ -r:x26reporting_event_name/ -i -o *.sas
gsar -s:x26delivery_name./ -r:x26delivery_name/ -i -o *.sas
gsar -s:x26execution_area./ -r:x26execution_area/ -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
```



Manual Inspection

- As a final step, the migrated programs must be thoroughly inspected, even when they seem to run without issues
- Depending on the level of standardisation adopted when writing the original program, only minor additional changes might be needed to make the program run correctly in the new environment
- Any command trying to open a shell to the underlying operating system, blocked in LSAF, must be analysed and substituted with a SAS function to make it platform independent



6

Conclusions



Conclusions

- As I stated in 2022, 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
- This time 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
- 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

KEY CONCEPTS APPLIED AND LESSONS LEARNED

- It always takes more time, effort, and people than you originally think
- Allow for thought time and trial-and-error when trailblazing
- Dedicate people to the project instead of trying to work on daily activities in parallel
- Bring in expertise when needed
- Lay out the project ground rules early and enforce them
- Establish a common project-oriented mindset among team members
- Fully integrate vendor and other external project team members
- Recognize and address the challenges of a geographically dispersed team
- Manage and prepare people for change



Questions?



Confidentiality Notice

This file is private and may contain confidential and proprietary information. If you have received this file in error, please notify us and remove it from your system and note that you must not copy, distribute or take any action in reliance on it. Any unauthorized use or disclosure of the contents of this file is not permitted and may be unlawful. AstraZeneca PLC, 1 Francis Crick Avenue, Cambridge Biomedical Campus, Cambridge, CB2 0AA, UK, T: +44(0)203 749 5000, www.astrazeneca.com

