

Manage TFLs Development in Life Science Analytics Framework 5.3 using SAS and R code

Stijn Rogiers, SAS Institute;
Jean-Marc Ferran, Qualiance;

ABSTRACT

TFL development and validation is often a labor-intensive task during the conduct of clinical trials. Collaboration between multiple programmers across the globe is often needed while oversight of progress can be difficult to tackle.

Life Science Analytics Framework (LSAF) 5.3 enables users to manage and execute both SAS and R programs in the same flexible and open platform. In addition, LSAF provides integrated features around versioning, eSignature and use of Workflows. These features coupled with an advanced API enable users to customize and streamline the definition and production of TFLs across studies and projects.

This presentation will feature, based on a prototype, how to specify TFLs, follow their development lifecycle, validation status using both SAS and R programs and keep control over development activities during the course of a study.

INTRODUCTION

The LSAF 4.x suite included powerful and simple features for versioning and eSignature of objects in the system. The combination of the two applied to programs, enable the users to define a simple Code Life-Cycle Management process based on the eSignatures of a programmer and a reviewer on a given version of a program. The implementation of such a process in LSAF 4.x was published and presented in 2018 [1] and also describes some of the possible pitfalls where users would not follow the defined process (e.g. programmer signs after validator, programmers and validators are the same, only validator eSigns, etc.) and is heavily dependent on the use of a “Programming Plan” in the form of an excel spreadsheet.

The LSAF 4.x suite already provided an advanced set of APIs [2] and a Workflow capability. The LSAF 5.x suite introduced a TFL module and more options for events triggering actions within Process Flow Definitions such as eSignature.

This paper and associated presentation explores how the initial Code Life Cycle Management [1] solely relying on eSignature and Versioning could be expanded to also utilize the new TFL module capability to replace the Programming Plan and advanced workflow features to automate and streamline further the review process and also capture key information (e.g. Reason for updating programs in Production).

A prototype was built to this effect considering two levels of validation (Self-Validation and Peer-Review) and different actions supporting the process (e.g. moving programs from a “Dev” to a “Prod” area and updating automatically programs attributes with Validation Status).

This paper introduces the key features in LSAF 5.3 to consider for Life-Cycle Management, presents the prototype and also explores the use of R within LSAF (new in 5.3 version, available since Nov 2020).

LSAF 5.3 TFL MODULE & PROCESS FLOWS

LSAF 5.3 provides a TFL module where:

- Each company can configure the metadata to be collected.
- TFL metadata can be managed at global level (Figure 1. **Configurable Global TFL Metadata**) and copied for each study.
- Study-level TFL metadata (Figure 2. **Study-Level TFL Metadata including User Assignments and Level of Validation**) can be adjusted and users assigned to the different tasks (programming, validation, peer-review, etc.) as well as Level of Validation.
- Workflows corresponding to each Level of Validation can be configured and assigned to each TFL (Figure 3. **Workflow assignment for different Levels of Validation** and Figure 4. **Process Flow Definitions at Study-level**).
- All metadata can be extracted manually or programmatically using APIs into SAS datasets for further consumption (Figure 5. **Export TFL Metadata using the LSAF APIs**).

Results	Enable Auto Flow	Section	Type	Identifier	Name	Description	program name	Title_1	Title_2	Title_3	Title_4	Title_5	Footnote_1	Footnote_2	Footnote_3	Footnote_4
<input type="checkbox"/>	No	1 - Subject Di...	Table	1	Subject Dis	All Randomized	SubjectDisposon.sas	Table 1.0	Subject Disp	Number (%)			[1] Randomized	[2] ITT Subjects	[3] Modified ITT	[4] Subject 148C
<input type="checkbox"/>	No	2 - Protocol V...	Table	2	Protocol Viol	Intent-to-Treat	ProtocolViolations.sas	Table 2.0	Protocol Viol	Number (%)			Note: Only one			
<input type="checkbox"/>	No	3 - PII Compl...	Table	3	PII Complia	Intent-to-Treat	PIICompliance.sas	Table 3.0	PII Comptiar	Number (%)			[1] First pill sh...	[2] Second pill s		
<input type="checkbox"/>	No	4 - Demograp...	Table	4	Demograph	Intent-to-Treat	DemographoCharact	Table 4.0	Demographs	Number (%)						
<input type="checkbox"/>	No	5 - Clinically...	Table	5	Clinical St	Intent-to-Treat	PreesingConditions	Table 5.0	Clinicaly Sig	Number (%)			Note: A clinica			
<input type="checkbox"/>	No	6 - Vital Signs...	Table	6	Vital Signs	Intent-to-Treat	VitalSignsFormations.s	Table 6.0	Vital Signs a							
<input type="checkbox"/>	No	7 - Summary...	Table	7	Summary of	Intent-to-Treat	PriorMedications.sas	Table 7.0	Summary of	By WHO-Dr	Number (%)		Note: Prior Med			
<input type="checkbox"/>	No	7 - Summary...	Table	7.1	Summary of	Intent-to-Treat	ConcomitantMedicatio	Table 7.1	Summary of	By WHO-Dr			Note: Concomit			
<input type="checkbox"/>	No	8 - Summary...	Table	8	Summary of	Intent-to-Treat	AdverseEvents.sas	Table 8.0	Summary of	By MedDRA			Note: Subjects			
<input type="checkbox"/>	No	8 - Summary...	Table	8.1	Summary of	Intent-to-Treat	DiscontinuationAdven	Table 8.1	Summary of	By MedDRA			Note: Subjects			
<input type="checkbox"/>	No	9 - Summary...	Table	9	Summary of	Intent-to-Treat	SeriousAdverseEvent	Table 9.0	Summary of	By MedDRA			Note: Subjects			
<input type="checkbox"/>	No	9 - Summary...	Table	9.1	Listing of In	Intent-to-Treat	SeriousAdverseEvent	Table 9.1	Listing of Inc				[1] Maximum Int	[2] Action Taken	[3] Drug Retire	[4] Outcome: Re
<input type="checkbox"/>	No	10 - Subject...	Table	10	Subject Rep	Intent-to-Treat	ReportedSymptoms.s	Table 10.0	Subject Repo	Number (%)			Note: Percenta			

Figure 1. Configurable Global TFL Metadata

Global Metadata for TFL includes:

- Type (Table/Figure/Listing)
- Identifier – Typically can be used for numbering of outputs
- Name/Description
- Title information
- Footnote information
- SAS or R Program Name associated with the output
- LSAF Job generating the output
- And other attributes or metadata that a company would like to capture for this purpose

It is possible to select TFL metadata from the global level to the study-level or copy from another similar study. When selecting TFL metadata for a given study, you can assign users and the Level of Validation.

All sections x

Standard type: Analysis Standard model: Analysis Standard Base standard name: ADaM

11 TLFs

<input type="checkbox"/>	Results	Enable Auto Flow	Section	Type	Identifier	Name	Description	SAS program name	Title_1	Title_2
<input type="checkbox"/>	2	Yes		Table	RD_Table_14-3.01	Table 14-3.01	Primary Endpoint An...	Table 14-3.01	Primary Endpoint Analysis...	- Summary at ...
<input type="checkbox"/>	1	Yes		Table	RD_Table_14-5.02	Table 14-5.02	Incidence of Treatm...	Table 14-5.02	Incidence of Treatment E...	by Treatment ...
<input checked="" type="checkbox"/>	1	Yes		Table	RD_Table_14-6.02	Table 14-6.02	Incidence of Treatm...	Table 14-6.02	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.03	Table 14-6.03	Incidence of Treatm...	Table 14-6.03	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.04	Table 14-6.04	Incidence of Treatm...	Table 14-6.04	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.05	Table 14-6.05	Incidence of Treatm...	Table 14-6.05	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.06	Table 14-6.06	Incidence of Treatm...	Table 14-6.06	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.07	Table 14-6.07	Incidence of Treatm...	Table 14-6.07	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.08	Table 14-6.08	Incidence of Treatm...	Table 14-6.08	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.09	Table 14-6.09	Incidence of Treatm...	Table 14-6.09	Incidence of Treatment E...	by Treatment ...
<input type="checkbox"/>	0	Yes		Table	RD_Table_14-6.10	Table 14-6.10	Secondary Endpoint...	Table 14-6.10	Secondary Endpoint Analy...	by Treatment ...

All sections x

Standard type: Analysis Standard model: Analysis Standard Base standard name: ADaM

11 TLFs

<input type="checkbox"/>	User Assignment 1	User Assignment 2	User Assignment 3	User Assignment 4	User Assignment 5	Output	Job
<input type="checkbox"/>	Jean-Marc Ferran (qua_jmf)	Stijn Rogiers (sbxsro)	Stijn Rogiers (sbxsro)				/PhUSE2020/Dru...
<input type="checkbox"/>	Stijn Rogiers (sbxsro)	Jean-Marc Ferran (qua_jmf)	Jean-Marc Ferran (qua_jmf)				/PhUSE2020/Dru...
<input type="checkbox"/>	Kevin Clark (keclar)	Adam LaManna (adlana)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Preetesh Parikh (prpari)	Ben Bocchicchio (bebocc)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Adam LaManna (adlana)	Sandeep Juneja (sajune)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Sandeep Juneja (sajune)	Kevin Clark (keclar)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)	Preetesh Parikh (prpari)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Preetesh Parikh (prpari)	Ben Bocchicchio (bebocc)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Adam LaManna (adlana)	Sandeep Juneja (sajune)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Adam LaManna (adlana)	Sandeep Juneja (sajune)	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...
<input type="checkbox"/>	Jean-Marc Ferran (qua_jmf)	Stijn Rogiers (sbxsro)	Sandeep Juneja (sajune)				/PhUSE2020/Dru...

All sections x

Standard type: Analysis Standard model: Analysis Standard Base standard name: ADaM

11 TLFs

<input type="checkbox"/>	User Assignment 3	User Assignment 4	User Assignment 5	Output	Job	Location	Due Date	Level of Validation	Utility Job 1
<input type="checkbox"/>	Stijn Rogiers (sbxsro)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Jean-Marc Ferran (qua_jmf)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			QC Review	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			Self-Verification	/PhUSE2020/Drug0...
<input type="checkbox"/>	Ben Bocchicchio (bebocc)				/PhUSE2020/Dru...			Self-Verification	/PhUSE2020/Drug0...
<input type="checkbox"/>	Sandeep Juneja (sajune)				/PhUSE2020/Dru...			Self-Verification	/PhUSE2020/Drug0...

Figure 2. Study-Level TFL Metadata including User Assignments and Level of Validation

LSAF uses the Business Process Model and Notation (BPMN) graphical representation standard. Different workflows can be developed and added to the system by the customer (no need for SAS involvement).

In our example, two workflows are created:

1. **SelfVerified:** Typically for an output that has low criticality and complexity and for which the programmer can validate its own output. The completion of the task is captured via eSignature.
2. **SignAndVersionAutomated:** Typically for an output requiring the review from another programmer. The output is considered validated when both the programmer and the validator have eSigned the same version of the program.

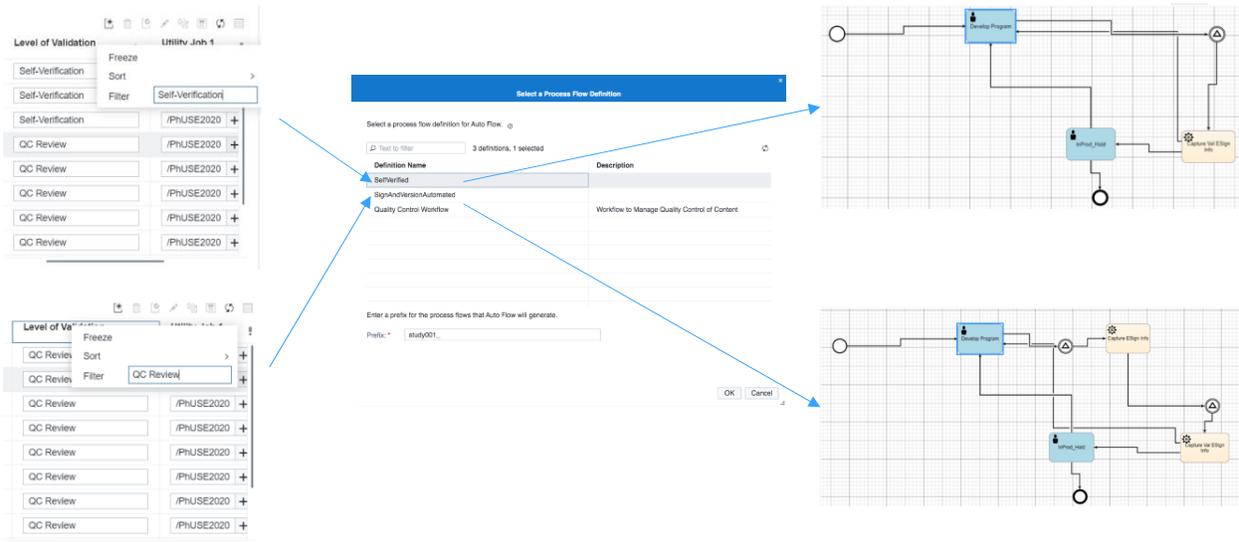


Figure 3. Workflow assignment for different Levels of Validation

The TFL module provides the ability to assign different workflows to each output deliverable according their Level of Validation. There is also the ability to have the assigned workflows created for each output in the system using the “Auto-Flow” feature.

Status	Name	Context	Creation Method	Process Flow Definition	Owner	Modified By	Date Modified
●	study001_Table 14-3.01	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Ben Bocchicchio	Ben Bocchicchio	Sep 8, 2020, 3:47 PM ...
●	study001_Table 14-5.02	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Ben Bocchicchio	Ben Bocchicchio	Sep 8, 2020, 3:47 PM ...
●	study001_Table 14-5.03	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Ben Bocchicchio	Ben Bocchicchio	Sep 8, 2020, 3:47 PM ...
●	study001_Table 14-5.04	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Ben Bocchicchio	Ben Bocchicchio	Sep 8, 2020, 3:47 PM ...
●	study001_Table 14-6.05	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Jean-Marc Ferran	Jean-Marc Ferran	Sep 14, 2020, 10:00 P...
●	study001_Table 14-6.06	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Ben Bocchicchio	Ben Bocchicchio	Sep 8, 2020, 3:47 PM ...
●	study001_Table 14-6.07	/PHUSE2020/Drug001/1...	TLF Auto Flow	SignAndVersionAutoma...	Ben Bocchicchio	Ben Bocchicchio	Sep 8, 2020, 3:47 PM ...
●	study001_Table 14-6.08	/PHUSE2020/Drug001/1...	TLF Auto Flow	SelfVerified	Jean-Marc Ferran	Jean-Marc Ferran	Sep 14, 2020, 9:59 PM ...
●	study001_Table 14-6.09	/PHUSE2020/Drug001/1...	TLF Auto Flow	SelfVerified	Jean-Marc Ferran	Jean-Marc Ferran	Sep 14, 2020, 9:27 PM ...
●	study001_Table 14-6.10	/PHUSE2020/Drug001/1...	TLF Auto Flow	SelfVerified	Jean-Marc Ferran	Jean-Marc Ferran	Sep 14, 2020, 9:59 PM ...

Figure 4. Process Flow Definitions at Study-level

Process Flow Definitions in LSAF 5.3 capture events such as eSignature and can move to a different state of the Workflow automatically as soon as a programmer or validator has eSigned a program in the system.

In addition, the LSAF APIs [2] enable the users to programmatically extract all metadata entered for a given study and consume them further for the actual coding of each analysis or for creating operational dashboards.

```

20
21 %lsaf_getstudyid(LSAF_PATH=&lsafstudy);
22 %put &_lsafStudyId_;
23
24 %lsaf_getstudystandards(
25   lsaf_studyid=%str(&_lsafStudyId_),
26   sas_dsname=work.lsafgetstudystandards
27 );
28
29 data _null_;
30   set work.lsafgetstudystandards end=end;
31   call symputx('modelid' || strip(put(_n_,8.)), modelid);
32   call symputx('standardname' || strip(put(_n_,8.)), name);
33   call symputx('basestandardname' || strip(put(_n_,8.)), basestandardname);
34   call symputx('basestandardversion' || strip(put(_n_,8.)), basestandardversion);
35   if end then call symputx('nstandards', _n_);
36 run;
37
38 %macro export_metadata(location);
39
40   %local i base_std;
41
42   %do i = 1 %to &nstandards;
43
44     %put &i &&modelid&i &&basestandardname&i &&standardname&i &&basestandardversion&i;
45
46     %let base_std = %lowcase(&&basestandardname&i);
47
48     %lsaf_exportstudydetails(
49       lsaf_studyid=%str(&_lsafStudyId_),
50       lsaf_location=&location,
51       lsaf_path=%str(&lsafstudy/Dev/metadata/&base_std/studydetails),
52       lsaf_overwrite=1
53     );
54
55   %if &location=WORKSPACE %then %do;
56     %lsaf_exportstudytablestows(
57       lsaf_studyid=%str(&_lsafStudyId_),
58       lsaf_basestandardname=&&basestandardname&i,
59       lsaf_modelid=%str(&&modelid&i)
60

```

Figure 5. Export TFL Metadata using the LSAF APIs

TFL LIFE-CYCLE MANAGEMENT IN LSAF, A PROTOTYPE

In order to illustrate the application of the TFL module and the use of workflows, a prototype was built using the two Process Flow Definitions described in the previous section and to which:

- Different actions were added to each state (Figure 7. **Automated Triggered Actions from eSignature**).
- A special state was added to capture when an output that has been validated must be sent to development again and where the reason is captured (Figure 8. **Workflow documents reasons for change after validation**).

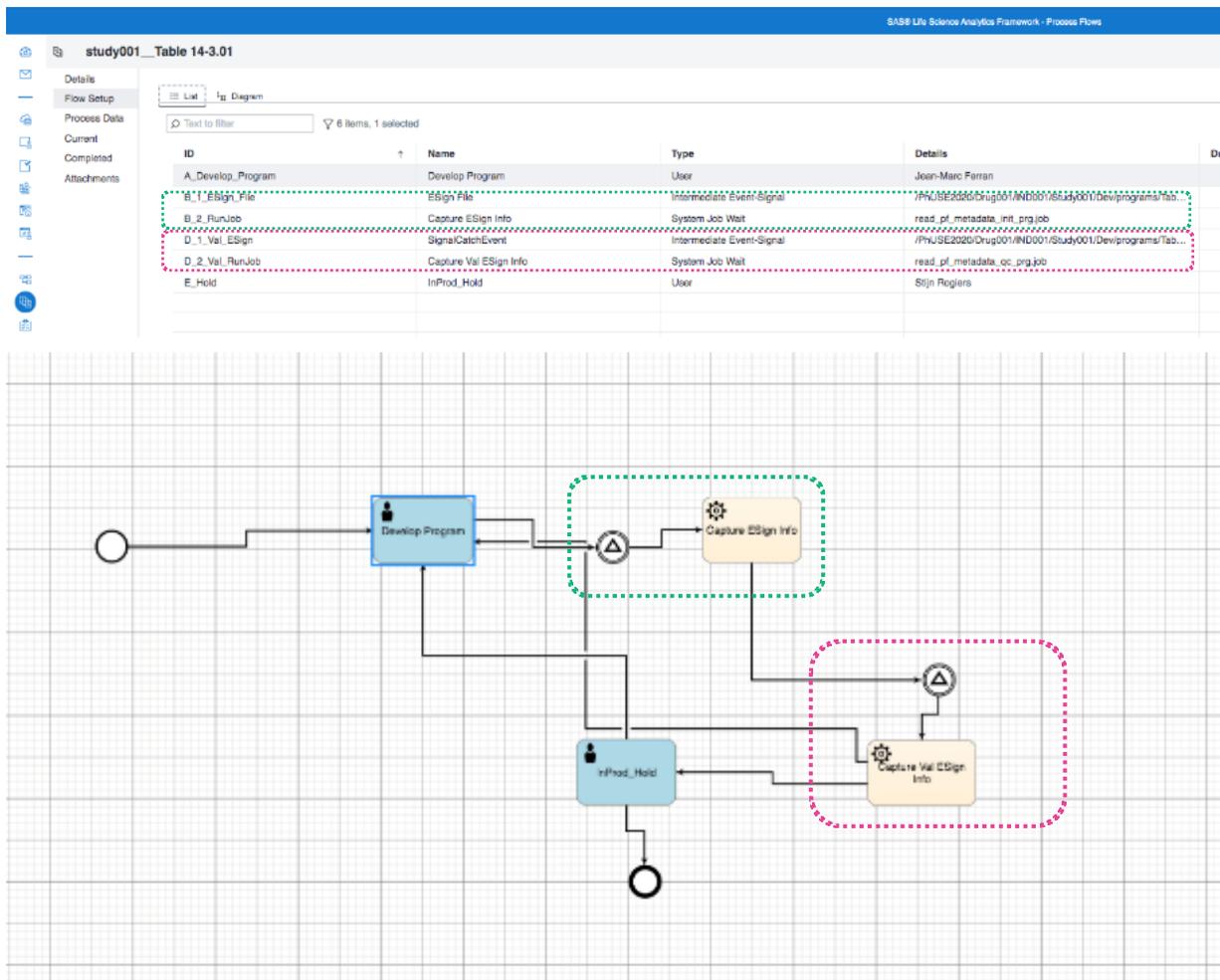


Figure 6. Process Flow Definitions for Life-Cycle Management using eSignature

In addition, programs were automatically moved from a “Dev” to a “Prod” area in the folder structure (and vice-versa) based on the validation status determined by the eSignature. The program description is also automatically updated with a code to indicate the Validation Status.

These automatically triggered actions are implemented via LSAF jobs using the LSAF APIs [2] for manipulating objects and managing objects attributes in the system. These jobs were associated with each state of the Process Flow Definitions and automatically triggered when a new eSignature was applied and detected in the system.

The two actions described here are only illustrative and it is possible to customize different types of actions in-line with each company Code Life-Cycle Management and Program Validation processes.

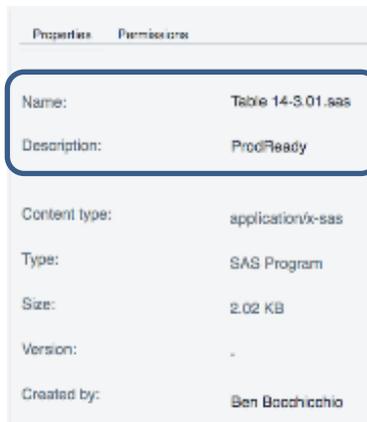
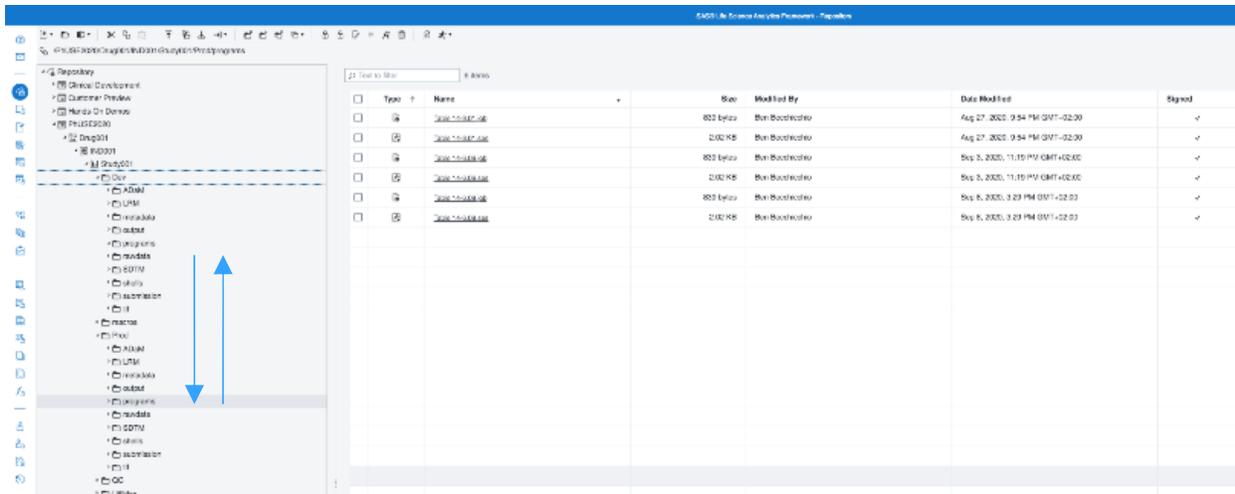


Figure 7. Automated Triggered Actions from eSignature

In the prototype, when an output has reached Production and must be demoted, the user must enter a reason (e.g. ReDefine Specifications). Such information is then saved in the Process Flow Definitions, can be read via APIs and reporting can be done to monitor further efficiency in the programming processes.

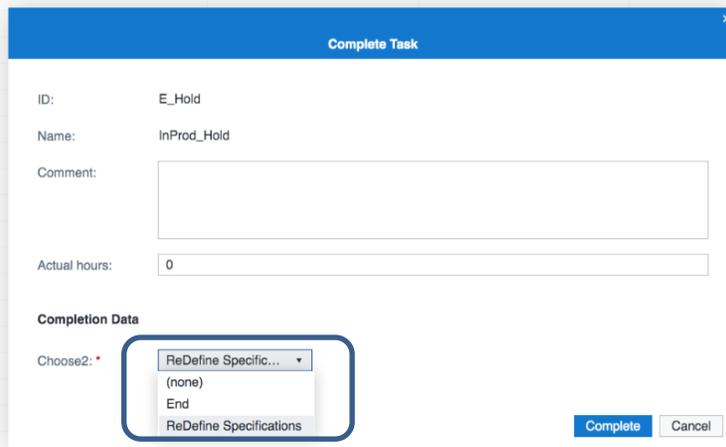


Figure 8. Workflow documents reasons for change after validation

All information captured in program attributes, process flow definitions or from the study-level metadata can be extracted using the LSAF APIs [2], consumed and used to produced metrics, dashboards or operation reports.

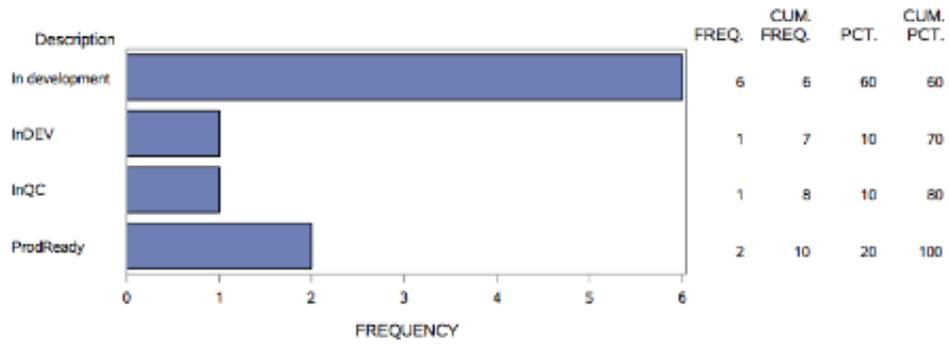


Figure 9. Operation Dashboard Example using Program Attributes

USING R IN LSAF 5.3 AND BEYOND

New in LSAF 5.3 is also the integration of R. Users can open a SAS session or an R session and code program(s) for analysis.

The Program Editor within LSAF supports the development of multiple programming languages including SAS and R.; provides color coding and produces an R log (.Rlog).

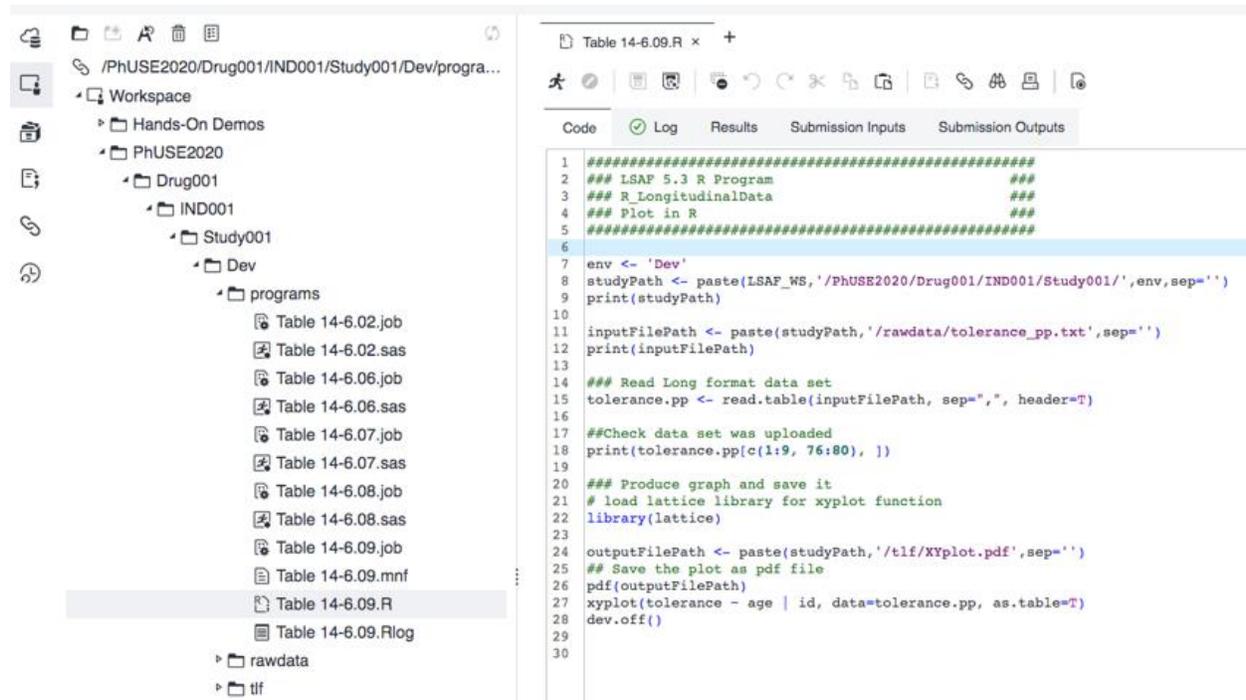


Figure 10. R Program Editor in LSAF

Important to note is that R code is managed the same way than SAS code in LSAF with similar processes using jobs and manifests to ensure traceability and reproducibility of each execution. Users are also able to have SAS and R programs combined within the same Job.

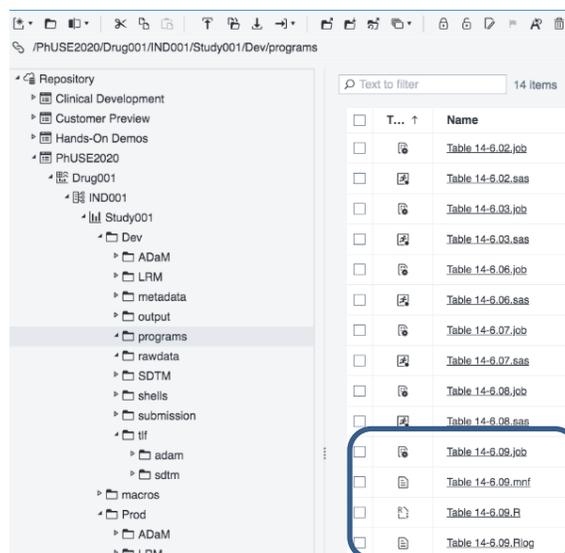


Figure 11. R Jobs, Logs and Manifests in LSAF 5.3

In Global-level or Study-level metadata, it is possible to assign SAS or R programs to different outputs and have them follow similar Code Life-Cycle Management using the same Workflows based on eSignature. This was verified in the prototype.

CONCLUSION

- **TLF module can be used for planning purpose AND to capture outputs metadata.**
 - ✓ TLF module can be configured to meet specific customer requirements
 - ✓ Export output metadata to a SAS dataset (using API or UI) to support the reporting of each TLF
- **It is possible to manage outputs supported by both SAS and R code**
 - ✓ Integrated within SAS Life Science Analytics Framework; no need to work outside system for dev. nor execution
 - ✓ Traceability and Reproducibility for both SAS and R programs
- **Setting up of Workflows can be automated using APIs and tailored to each company's internal process**
 - ✓ Workflow can be triggered on eSign activities & activate next steps accordingly (e.g. move from DEV to PROD area)
 - ✓ It is possible to document after the validation state is reached any reason for change (e.g. change of specs)

REFERENCES

[1]: Ferran, Jean-Marc. 2018. "Managing Programs Development Life-Cycles in SAS LSAF". Proceedings of the PHUSE 2018 US Connect Conference, Raleigh, NC. Available at <https://www.lexjansen.com/phuse-us/2018/dh/DH09.pdf>

[2]: "Life Science Analytics Framework 5.1.x / 5.2.x / 5.3 - Access to available versions of the Java API client and Macro API download components". Accessed May 13, 2021. <https://support.sas.com/kb/57/862.html>.

ACKNOWLEDGMENTS

We would like to thank Ben Bocchicchio and Sandeep Juneja for developing the prototype that is featured in this paper.

CONTACT INFORMATION

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

Stijn Rogiers
SAS Institute
stijn.rogiers@sas.com
<http://www.sas.com>
<https://www.linkedin.com/in/stijnrogiers/>

Jean-Marc Ferran
Qualiance
JMF@qualiance.dk
<http://www.qualiance.dk>
<https://www.linkedin.com/in/jeanmarcferran/>

Any brand and product names are trademarks of their respective companies.