



Validating outsourced programming tasks by comparing SAS datasets without PROC COMPARE

PHUSE EU Connect Birmingham, 7 November 2023

Herman Ament, Danone Nutricia Research



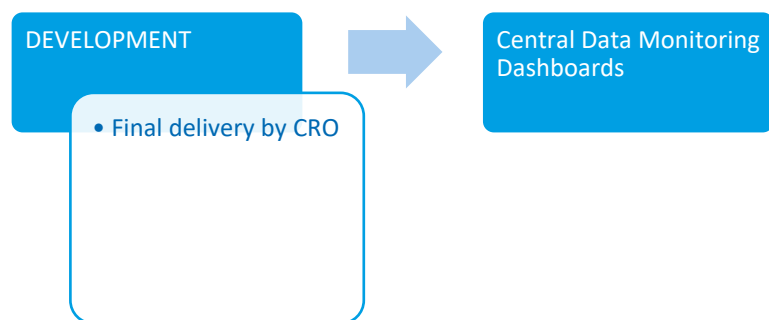
AGENDA

- Introduction
- Programming lifecycle and environment
- About hashing or checksums
- Comparing datasets using hashing
- Reasons of a mismatch
- Final remarks and conclusion

INTRODUCTION

- SAS programs that convert EDC data to SDTM SAS Datasets are the key deliverables of the CRO for Danone Nutricia Research (DNR). SAS programs need to be run in the LSAF environment of DNR
- DNR standard macros, metadata and source (dummy) data are shared with the CRO
- Validated programs developed by CRO are delivered to DNR.
- Programs are uploaded and executed in the DNR LSAF environment
- Result of executing the programs at DNR in LSAF should be that SDTM SAS datasets are identical at DNR
- **Instead of comparing all SDTM SAS datasets individually, one special (hash) dataset is used to compare all SDTM SAS datasets at the CRO versus the SAS datasets created at DNR**

LIFECYCLE OF THE SDTM PROGRAMS OF A STUDY



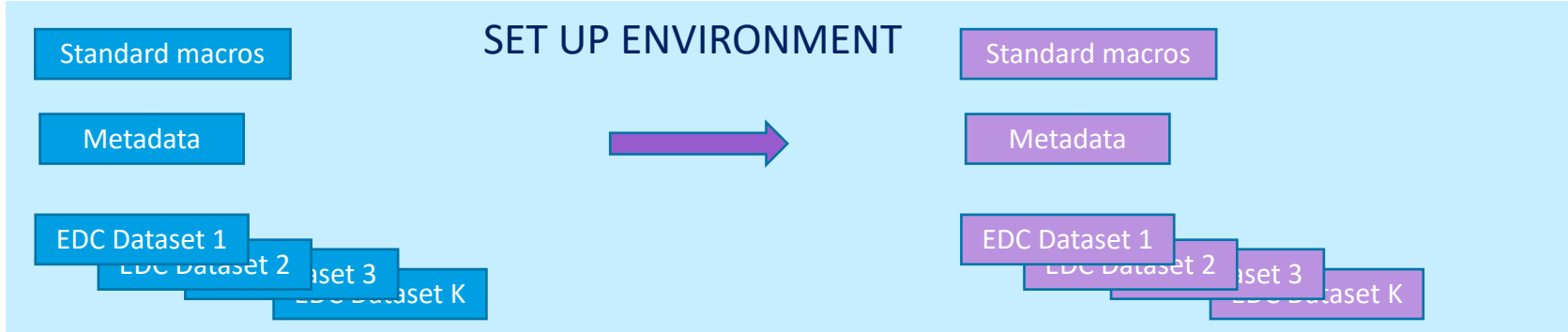
LSAF DNR

SAS CRO

Standards

Standard Metadata version x

(Dummy) Source data



SDTM SAS Programs

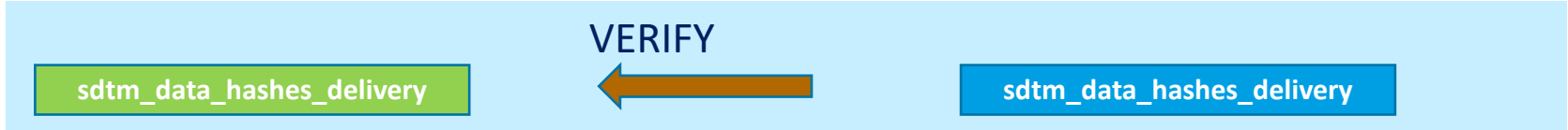
Study specific macros



SDTM SAS datasets (CRO)

Dataset 1
Dataset 2
Dataset 3
Dataset N

Hash SAS dataset



SDTM SAS datasets (LSAF)

Dataset 1
Dataset 2
Dataset 3
Dataset N

Compare hash SAS dataset

Compare SAS datasets hashes <date>.pdf

One hash dataset to verify that all SDTM SAS datasets are the same
-> No copy of SDTM SAS datasets in LSAF

ABOUT HASHING OR CHECKSUMS

- From SAS Help:
“Hashing is the transformation of data (known as a message) into a **short fixed-length value** (known as a digest) that **represents the original string**. Hashing is used to index and retrieve items in a database because it is faster to find the item using the hashed key than using the original value.
Hashing functions are useful when you are developing shell scripts for software installation, **file comparison**, and detection of file corruption and tampering”
- Algorithms supported by SAS: **CRC32**, MD5, SHA1, SHA256, SHA384, SHA512
- In general, the larger the hash, the less chance that the same hash is created from different original strings
- DNR has chosen to use the CRC32 method. The hash string is **short**, several hashes can be displayed easily next to each other, and still the chance on a collision is very low ($1:10^{10}$)
- This is very acceptable for comparing for our purpose: To compare SAS datasets (HASHING_FILE ('CRC32', ..))

Method	Hash (checksum) in hexadecimal presentation (Digest)	Display Length
CRC32	4D170E0E	8 ✓
MD5	C1A5298F939E87E8F962A5EDFC206918	32
SHA1	94DD9E08C129C785F7F256E82FBE0A30E6D1AE40	40
SHA256	3639EFCD08ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8	64

HOW TO USE HASHING FOR COMPARING SAS DATASETS

- Can we hash original SAS Dataset file **✗**
SAS datasets contain also metadata about the SAS environment, creation and modified datetime stamp
Therefore, each SAS dataset is different
- To mimic PROC COMPARE:
 - Compare the values
 - Compare attributes of the variables (format, informat, type, position, sorted)
 - Compare label of the dataset
- Let's have a look how this works ...

WOW - AT THE CRO

- Programs are finalised and executed
- Deliverables for Danone Nutricia
 - SAS Programs
 - Study specific macros

... and...

- Delivery of hash dataset (final step)
 - Run “Create SAS datasets hashes” (standard macro %hash_set)
 - QC SAS dataset: **sdtm_data_hashes_delivery**

Example of the hash dataset for SDTM datasets MH and ML

memname	hash_data	hash_set	hash_var	hash_oth	sort	nobs	nvar	encoding	memlabel	hash_metho	datarepname
MH	89AD9EB9	0BAB5DA5	C0DDD4C0	A335C843	N	7	32	utf-8 Unicode...	SDTM domain MH...	CRC32	WINDOWS_64 ... \$
ML	F6E4F18C	D517BEAF	5A8DE95B	601C366C	N	44	17	utf-8 Unicode...	SDTM domain ML ...	CRC32	WINDOWS_64 ... \$

- Also delivered: QC Documentation

WOW - AT DANONE NUTRICIA (1/2)

- In LSAF production
 - Add/Update SAS programs + study specific macros in LSAF
 - Run all programs
 - Result in LSAF: **SDTM SAS Datasets**

... and...

- Compare hash datasets (final step, we hope 🤔)
 - Run **Compare SAS datasets hashes.job** (global macro %hash_set)
 - Result PDF file: **Compare SAS datasets hashes <date>.pdf**
- Review PDF

WOW - AT DANONE NUTRICIA (2/2)

- Review PDF, cells highlighted with smart traffic light colors
- All cells green, conclusion is
Datasets are exactly the same at both sides ✓
- No further action needed

Compare hashes result of delivery 2023-05-23

Result	Sort before creating hash data	Source	Member Name	Hash of data	Hash of memlabel	Hash of attributes name, label, type	Hash of other attributes (format, informat, sorted, length, varnum)
All Match	N	LSAF	MH	89AD9EB9	0BAB5DA5	C0DDD4C0	A335C843
	N	Delivered	MH	89AD9EB9	0BAB5DA5	C0DDD4C0	A335C843
All Match	Y	LSAF	MH	6E1A5C8C	0BAB5DA5	C0DDD4C0	A335C843
	Y	Delivered	MH	6E1A5C8C	0BAB5DA5	C0DDD4C0	A335C843
All Match	N	LSAF	ML	F6E4F18C	D517BEAF	5A8DE95B	601C366C
	N	Delivered	ML	F6E4F18C	D517BEAF	5A8DE95B	601C366C
All Match	Y	LSAF	ML	C344131E	D517BEAF	5A8DE95B	601C366C
	Y	Delivered	ML	C344131E	D517BEAF	5A8DE95B	601C366C

DETAILS ABOUT THE HASH (CHECKSUM) VARIABLES IN THE HASH DATASET (1/2)¹¹

- Comparison is done
 - All contents
 - Label dataset
 - Attributes: name, label type
 - Other attributes: formats, informats, sorted, length and varnum

- Comparison on not-sorted data and sorted data



Result	Sort before creating hash data	Source	Member Name	Hash of data	Hash of memlabel	Hash of attributes name, label, type	Hash of other attributes (format, informat, sorted, length, varnum)
All Match	N	LSAF	MH	89AD9EB9	0BAB5DA5	C0DDD4C0	A335C843
	N	Delivered	MH	89AD9EB9	0BAB5DA5	C0DDD4C0	A335C843
All Match	Y	LSAF	MH	6E1A5C8C	0BAB5DA5	C0DDD4C0	A335C843
	Y	Delivered	MH	6E1A5C8C	0BAB5DA5	C0DDD4C0	A335C843
All Match	N	LSAF	ML	F6E4F18C	D517BEAF	5A8DE95B	601C366C
	N	Delivered	ML	F6E4F18C	D517BEAF	5A8DE95B	601C366C
All Match	Y	LSAF	ML	C344131E	D517BEAF	5A8DE95B	601C366C
	Y	Delivered	ML	C344131E	D517BEAF	5A8DE95B	601C366C

DETAILS ABOUT THE HASH (CHECKSUM) VARIABLES IN THE HASH DATASET (1/2)¹²

- For each dataset
 - For each component (data values, label dataset, attributes and other attributes) a temporary flat file is created
 - Of each flat file a hash is created (function HASHING_FILE)
 - Result is one row with 4 hash variables
 - This is repeated before and after sort. BY statement is a list of variables sorted by case insensitive names
 - This is repeated for each dataset
- Example of a flat file of the metadata for the attributes Label (all missing), Name, Type (1 or 2) of the variables in SASHELP.CLASS

```
1# #Age#1## 2# #Height#1## 3# #Name#2## 4# #Sex#2## 5# #Weight#1###
```

- Example of the first record in the flat file from SASHELP.CLASS: Age, Height, Name, Sex and Weight for “Alfred”.

[illegible]

- Note: to ensure an exact match on numeric values the internal presentation is used (format binary64.)

REASONS OF A MISMATCH

- Review PDF, one or more cells red/orange
Datasets are not exactly the same at both sides 
or one source is missing

Result	Sort before creating hash data	Source	Member Name	Hash of data	Hash of memlabel	Hash of attributes name, label, type	Hash of other attributes (format, informat, sorted, length, varnum)
Data differ	N	LSAF	MH	9CA3385C	0BAB5DA5	C0DDD4C0	0C040DC3
	N	Delivered	MH	338D1289	0BAB5DA5	C0DDD4C0	913278C1
Data differ	Y	LSAF	MH	9CA3385C	0BAB5DA5	C0DDD4C0	0C040DC3
	Y	Delivered	MH	FE5867C0	0BAB5DA5	C0DDD4C0	913278C1

- Further investigation is needed
- Further investigation may depend on the number of SDTM SAS dataset that differ
- Possible reasons:
 - Different EDC source data, resulting in different numbers of observations and thus differences in hash of the data
-> many datasets are unequal
 - Difference in the set-up (STUDYID) , same number of observations -> many datasets unequal
 - Metadata not applied in a program, e.g. sorting not applied -> hash of other attributes
 - Different version of the DNR metadata: (a few or many differences)
 - Difference in the labels of variables
 - Difference in the number of variables
 - Difference in the data
 - ...
 - Different SAS system options, for example OPTIONS MISSING =
 - ...

FINAL REMARKS

- SAS Macros:
 - %hash_set hashes of one SAS dataset
 - %hash_lib hashes of all SAS datasets in one library
 - %hash_readback create from the written flat files a SAS dataset
 - %hash_verify hash a SAS dataset, read back and compare
- %hash_verify
Validation via reverse engineering : create original dataset back from the component files ✓
- Validation by comparing hashes at CRO and at DNR of several SASHELP SAS datasets: CLASS, BASEBALL, HEART in LSAF (Unix) and CRO Windows Platform. ✓

CONCLUSION

- Very convenient to ensure that programs create identical SDTM SAS datasets at both sides
- Only one small dataset to verify that all created SAS SDTM datasets are the same
- Message @SAS®
 - Consider to add hash values as metadata to SAS Datasets
 - To be retrieved with PROC CONTENTS / from DICTIONARY.TABLES
 - To be used to compare SAS datasets without PROC COMPARE
- Further reading PHUSE paper “EU PHUSE 2023 - SM03”

Contact: Herman Ament, Sr. Stat. Programmer. herman.ament@danone.com

Thank you