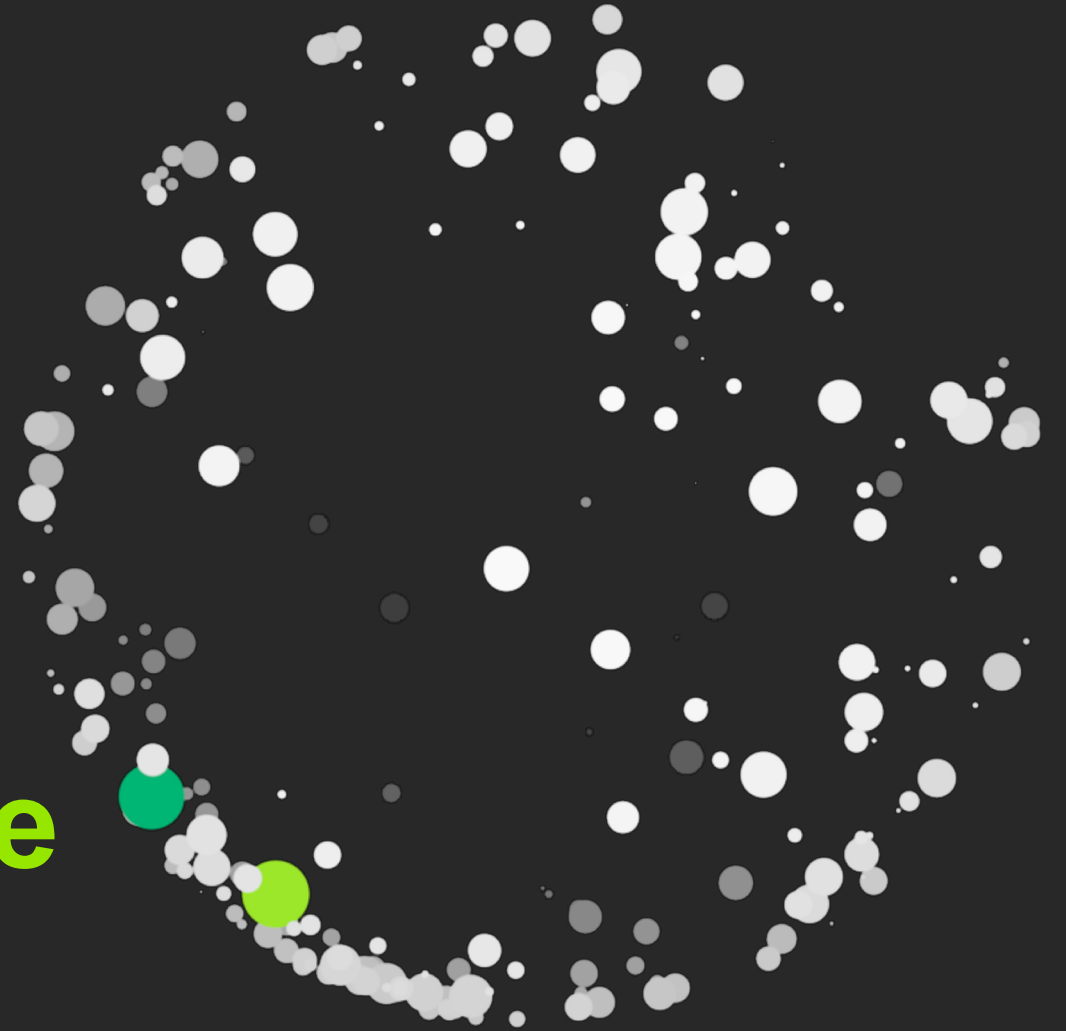
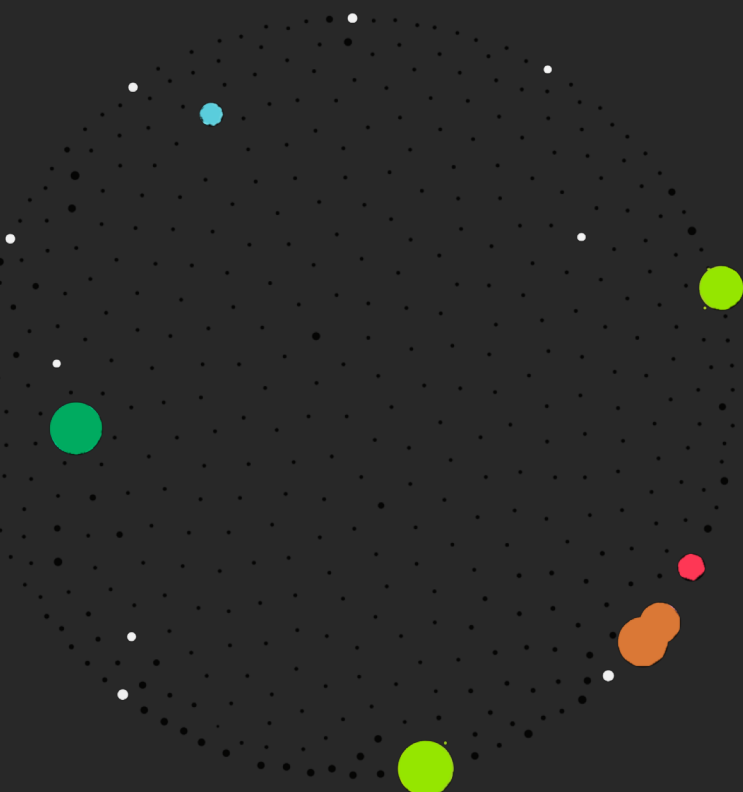


Exchange of SAS programs between local/server based SAS and the SAS Life Science Analytical Framework



INTRODUCTION



SAS LSAF has been introduced as a central data repository

Current SAS dev/prod environment is local/server based

Central data repository has been expanded

Local/server based SAS as development environment,
SAS LSAF as validation/production environment

Global SAS macros must be available in both
environments

Framework must ensure that the SAS code was not change
during exchange

WAYS OF EXCHANGING SAS DATA AND FILES

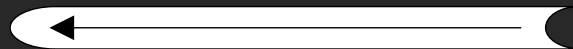
WEBDAV

Local Client

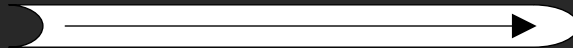


SAS
DDDC

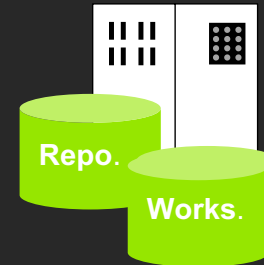
Download



Upload



SAS LSAF



LIMITATIONS

- Official only supported for Windows OS
- Download is very slow for large files
- File size limitation

MACRO/JAVA API

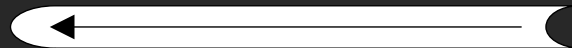
Local Client



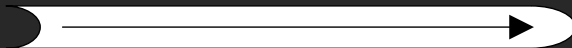
SAS
Session

Java
Application

Download



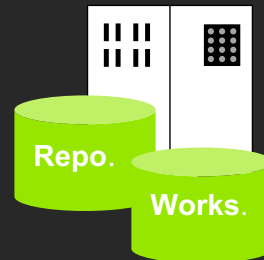
Upload



System Management



SAS LSAF



LIMITATIONS / MACROS

- LSAF_LOGIN must run before other macros can be used
- Downloaded content is always stored on a drive before it can be processed further

WAYS OF EXCHANGING SAS DATA AND FILES

FILEHANDLER

Local Client

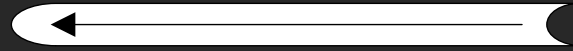


SAS
Session

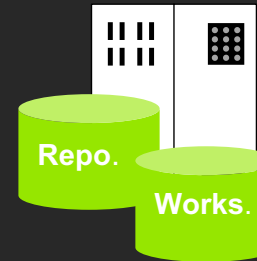
R
Session

Python
Session

Download



SAS LSAF



LIMITATIONS

- only download
- password in clear text (applies for R)

```
1
2 /* SAS example */
3 filename out "ae.xpt";
4
5 proc http
6   method='get'
7   url="https://<subdomain>.ondemand.sas.com/lsaf/filehandler/sdd:/<repository-path>/sdtm/xpt/ae.xpt"
8   webusername="&sdduser."
9   webpassword="&_sddpw."
10  out=out;
11 run;
12
13 /* R example */
14 cdwuser <- "<LSAF username>"
15 cdwpasswd <- "<LSAF password>"
16 download.file(url="https://cdwuser:cdwpasswd@<subdomain>.ondemand.sas.com/lsaf/filehandler/sdd:/<repository-path>/sdtm/xpt/ae.xpt",destfile='c:/tmp/ae.xpt')
17
```

PROBLEM

A SAS program, which should run in both environments, must use the same SAS code and the same macros. This means that macros and catalogs must be present in both environments. A code change in just one valid environment is not allowed. Furthermore all macros must be able to determine in which environment they are submitted.

MAIN TASK

Replace the SAS Drug Development Desktop Connection (SAS DDDC) tool and create system-independent SAS macros.

IMPLEMENTATION PHASE 1

Requirements

- 1 Login information is not displayed in the SAS log and macros do not need a separate login macro
- 2 A macro which returns the information in which environment the code is executed (supportive macro)
- 3 Macros for SAS data and file download
- 4 Macro for SAS data and file upload

We developed some more supportive macros for the interaction between the local/server based SAS and the SAS Life Science Analytics Framework, but these are not relevant for the here described solution.

SECURE SAS LIFE SCIENCE ANALYTICAL FRAMEWORK LOGIN

NOTE:

The SAS macro SASDRUGDEV_SESSIONLOGIN is used, which is described in the SAS Problem Note 49562

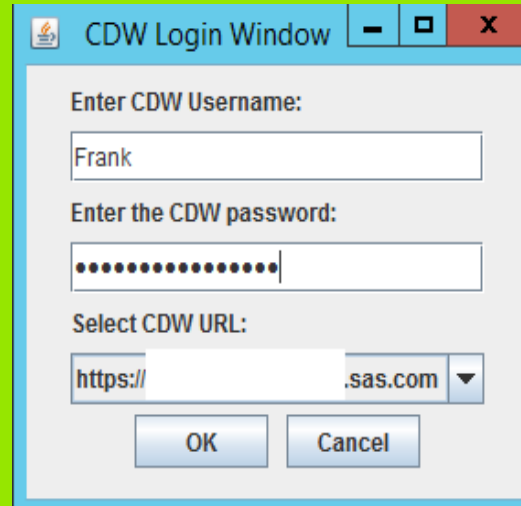
MACRO REQUIRES A TEXT FILE:

sddcreds.txt - Notepad

File Edit Format View Help

```
sdduser=<user>
sddpw=<password>
sddurl=https://<environment>.sas.com
```

JAVA PROGRAM WHICH USE THE APIs:



JAVA PROGRAM OUTCOME:

sddcreds.txt - Notepad

File Edit Format View Help

```
sdduser=<user>
sddpw={SAS001}XkZtNjgtt6Y0NyE=
sddurl=https://<environment>.sas.com
```

NEW LOGIN MACRO GRTCDW_SESSIONLOGIN - EXTRACT:

```
1
2 /* The supportive macro grtcdw_whereami controls in which
3 environment the macro is executed: */
4 %IF %grtcdw_whereami=windows %THEN %DO;
5 /* Active sessions have to be stopped before you can log
6 in again. Global macro variables must be deleted. */
7 %IF %SYMEXIST(sddsession) %THEN %DO;
8 %sasdrugdev_logout;
9 %SYMDEL sddsession sddcredfile;
10 %PUT Deleting macro variable SDDSESSION from scope GLOBAL.;
11 %PUT Deleting macro variable SDDCREDFILE from scope GLOBAL.;
12 %END;
13 /* Create a macro variable that contains the path to the login file.
14 Call the macro SASDRUGDEV_SESSIONLOGIN. Create the macro variable
15 sddsession and set the value active. */
16 %GLOBAL sddcredfile sddsession;
17 %LET sddcredfile = C:\Users\&sysuserid\sddcreds.txt;
18 %sasdrugdev_sessionlogin();
19 FILENAME _all_ CLEAR;
20 %LET sddsession=active;
21 %END;
22
```

THE SUPPORTIVE MACRO: WHERE AM I

A simple SAS macro, which just returns a macro variable containing the information in which environment the code is submitted.

In the SAS Life Science Analytics Framework the macro returns even, if it is submitted in the workspace or in the repository.

POSSIBLE RESULTS:

```
1
2 %LET _grtcdw_whereami=workspace;
3 %LET _grtcdw_whereami=repository;
4 %LET _grtcdw_whereami=windows;
5
```

MACROS: FOR DATA AND FILE DOWNLOAD

GRTCDW_DOWNLOAD:

- SAS related files like SAS7BDAT, CPORT and XPORT can be downloaded
- datasets are only stored in the temporary SAS session folder
- XPORT and CPORT are automatically extracted
- Different versions can be downloaded
- No additional login is required
- ZIP compressed download
- Environment check, WIN= download, LSAF = sync.

EXAMPLE:

```
1
2 /* All datasets - new library */
3 %grtc_dw_download(cdwpath=<path>/sdtm/xpt,
4   dsetlist=_ALL_,
5   libout=new_lib);
6
7 /* Four datasets */
8 %grtc_dw_download(cdwpath=<path>/sdtm/xpt,
9   dsetlist=ex dm sv ae,
10  libout=work);
11
12 /* Two datasets - different versions */
13 %grtc_dw_download(cdwpath=<path>/sdtm/xpt,
14   dsetlist=dm ae, dsetvers=2.0 3.0,
15   libout=work);
16
```

GRTCDW_FILEDOWNLOAD:

- EXCEL, PDF, WORD, ASCII, XML and other file type can be downloaded
- files are only stored in the temporary SAS session folder
- Different versions can be downloaded
- No additional login is required
- Environment check, WIN= download, LSAF = sync.

EXAMPLE:

```
1
2 /* Three files */
3 %grtc_dw_filedownload(cdwpath=<path>/sdtm/xpt,
4   filelist=file1.pdf file2.xml file3.xlsx);
5
6 /* Two files - different versions */
7 %grtc_dw_filedownload(cdwpath=<path>/sdtm/xpt,
8   filelist=file1.pdf file3.xlsx,
9   filevers=2.0 1.0);
10
```

MACRO: FOR DATA AND FILE UPLOAD

GRTCDW_UPLOAD:

- SAS related files like SAS7BDAT, CPORT and XPORT can be uploaded
- Files can be uploaded from a library or from a local directory
- Subfolders are automatically created in the data upload folder
- No additional login is required
- ZIP compressed upload
- Environment check, WIN = upload, LSAF = do nothing.

EXAMPLE:

```
1
2  /* Upload one file from the work library */
3  %grtc dw_upload(cdwpath=/
```

IMPLEMENTATION PHASE 2

Requirements

- 1 Implement in LSAF a shared macro library
- 2 Macro which collects all used inputs and outputs within the program(s)
- 3 Macros to create and execute a SAS LSAF job
- 4 Macro to create a ZIP file, which contains the required folder structure and all programming files

IMPLEMENT IN SAS LSAF A SHARED MACRO LIBRARY

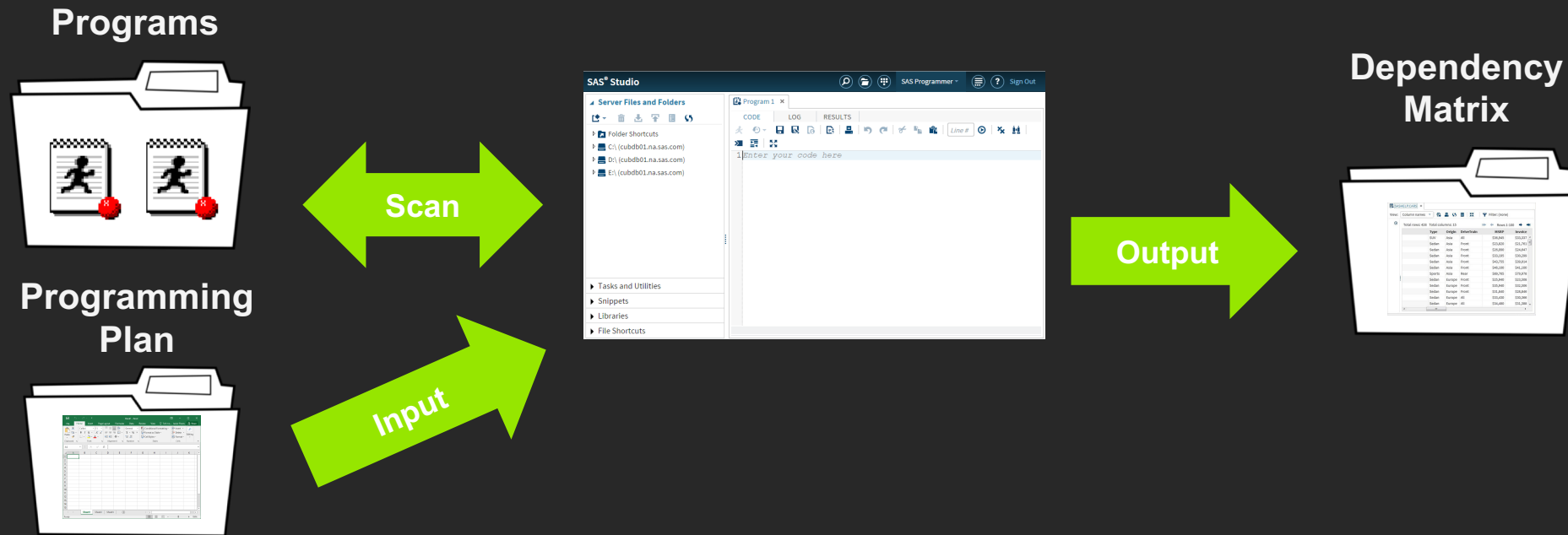
General SAS LSAF job approach



SAS LSAF job with the shared macro library concept



MACRO: FOR THE INPUT AND OUTPUT COLLECTION

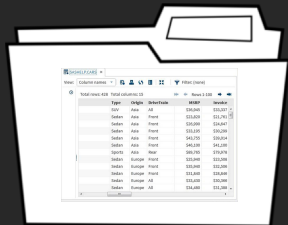


DEPMACROS:

- collects all programs belonging to the current batch as listed in the programming plan
- it determines, which inputs and outputs are used within the programs
- two macro types are distinguished: generic (global) macros and study specific macros

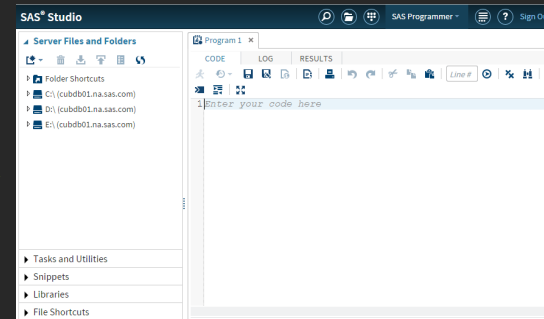
MACRO: TO CREATE A SAS LSAF JOB

Dependency
Matrix



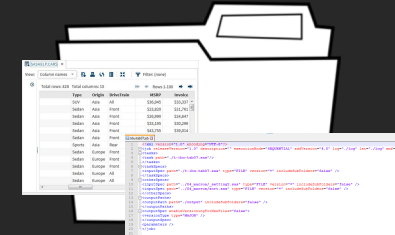
Program	Dependencies
Program 1	Program 2
Program 2	Program 3
Program 3	Program 4
Program 4	Program 5
Program 5	Program 6
Program 6	Program 7
Program 7	Program 8
Program 8	Program 9
Program 9	Program 10
Program 10	Program 11
Program 11	Program 12
Program 12	Program 13
Program 13	Program 14
Program 14	Program 15
Program 15	Program 16
Program 16	Program 17
Program 17	Program 18
Program 18	Program 19
Program 19	Program 20
Program 20	Program 21
Program 21	Program 22
Program 22	Program 23
Program 23	Program 24
Program 24	Program 25
Program 25	Program 26
Program 26	Program 27
Program 27	Program 28
Program 28	Program 29
Program 29	Program 30
Program 30	Program 31
Program 31	Program 32
Program 32	Program 33
Program 33	Program 34
Program 34	Program 35
Program 35	Program 36
Program 36	Program 37
Program 37	Program 38
Program 38	Program 39
Program 39	Program 40
Program 40	Program 41
Program 41	Program 42
Program 42	Program 43
Program 43	Program 44
Program 44	Program 45
Program 45	Program 46
Program 46	Program 47
Program 47	Program 48
Program 48	Program 49
Program 49	Program 50
Program 50	Program 51
Program 51	Program 52
Program 52	Program 53
Program 53	Program 54
Program 54	Program 55
Program 55	Program 56
Program 56	Program 57
Program 57	Program 58
Program 58	Program 59
Program 59	Program 60
Program 60	Program 61
Program 61	Program 62
Program 62	Program 63
Program 63	Program 64
Program 64	Program 65
Program 65	Program 66
Program 66	Program 67
Program 67	Program 68
Program 68	Program 69
Program 69	Program 70
Program 70	Program 71
Program 71	Program 72
Program 72	Program 73
Program 73	Program 74
Program 74	Program 75
Program 75	Program 76
Program 76	Program 77
Program 77	Program 78
Program 78	Program 79
Program 79	Program 80
Program 80	Program 81
Program 81	Program 82
Program 82	Program 83
Program 83	Program 84
Program 84	Program 85
Program 85	Program 86
Program 86	Program 87
Program 87	Program 88
Program 88	Program 89
Program 89	Program 90
Program 90	Program 91
Program 91	Program 92
Program 92	Program 93
Program 93	Program 94
Program 94	Program 95
Program 95	Program 96
Program 96	Program 97
Program 97	Program 98
Program 98	Program 99
Program 99	Program 100

Input



Output

LSAF Job/
Metadata



Program	Dependencies
Program 1	Program 2
Program 2	Program 3
Program 3	Program 4
Program 4	Program 5
Program 5	Program 6
Program 6	Program 7
Program 7	Program 8
Program 8	Program 9
Program 9	Program 10
Program 10	Program 11
Program 11	Program 12
Program 12	Program 13
Program 13	Program 14
Program 14	Program 15
Program 15	Program 16
Program 16	Program 17
Program 17	Program 18
Program 18	Program 19
Program 19	Program 20
Program 20	Program 21
Program 21	Program 22
Program 22	Program 23
Program 23	Program 24
Program 24	Program 25
Program 25	Program 26
Program 26	Program 27
Program 27	Program 28
Program 28	Program 29
Program 29	Program 30
Program 30	Program 31
Program 31	Program 32
Program 32	Program 33
Program 33	Program 34
Program 34	Program 35
Program 35	Program 36
Program 36	Program 37
Program 37	Program 38
Program 38	Program 39
Program 39	Program 40
Program 40	Program 41
Program 41	Program 42
Program 42	Program 43
Program 43	Program 44
Program 44	Program 45
Program 45	Program 46
Program 46	Program 47
Program 47	Program 48
Program 48	Program 49
Program 49	Program 50
Program 50	Program 51
Program 51	Program 52
Program 52	Program 53
Program 53	Program 54
Program 54	Program 55
Program 55	Program 56
Program 56	Program 57
Program 57	Program 58
Program 58	Program 59
Program 59	Program 60
Program 60	Program 61
Program 61	Program 62
Program 62	Program 63
Program 63	Program 64
Program 64	Program 65
Program 65	Program 66
Program 66	Program 67
Program 67	Program 68
Program 68	Program 69
Program 69	Program 70
Program 70	Program 71
Program 71	Program 72
Program 72	Program 73
Program 73	Program 74
Program 74	Program 75
Program 75	Program 76
Program 76	Program 77
Program 77	Program 78
Program 78	Program 79
Program 79	Program 80
Program 80	Program 81
Program 81	Program 82
Program 82	Program 83
Program 83	Program 84
Program 84	Program 85
Program 85	Program 86
Program 86	Program 87
Program 87	Program 88
Program 88	Program 89
Program 89	Program 90
Program 90	Program 91
Program 91	Program 92
Program 92	Program 93
Program 93	Program 94
Program 94	Program 95
Program 95	Program 96
Program 96	Program 97
Program 97	Program 98
Program 98	Program 99
Program 99	Program 100

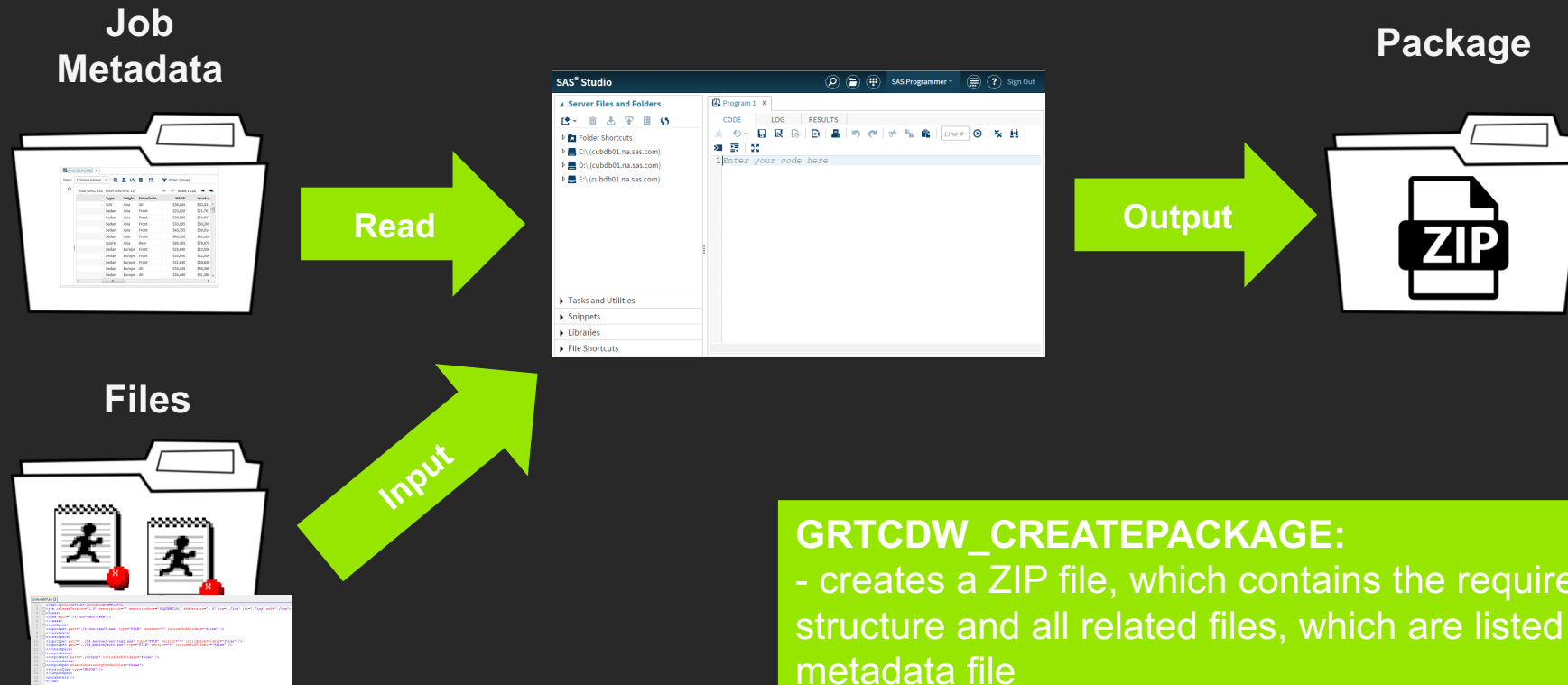
GRTCDW_CREATEJOB:

- creates from the program(s) which are listed in the dependency matrix dataset SAS LSAF job(s)
- creates one job related metadata dataset per program

EXAMPLE:

```
1
2 /* Create SAS LSAF job */
3 %grtc dw_creatjob(mdspath=C:\dev\metadata,
4     mdds=depmatrix,
5     mdprg=t-ibu-tab07);
6
```

MACRO: TO CREATE A PROGRAM PACKAGE

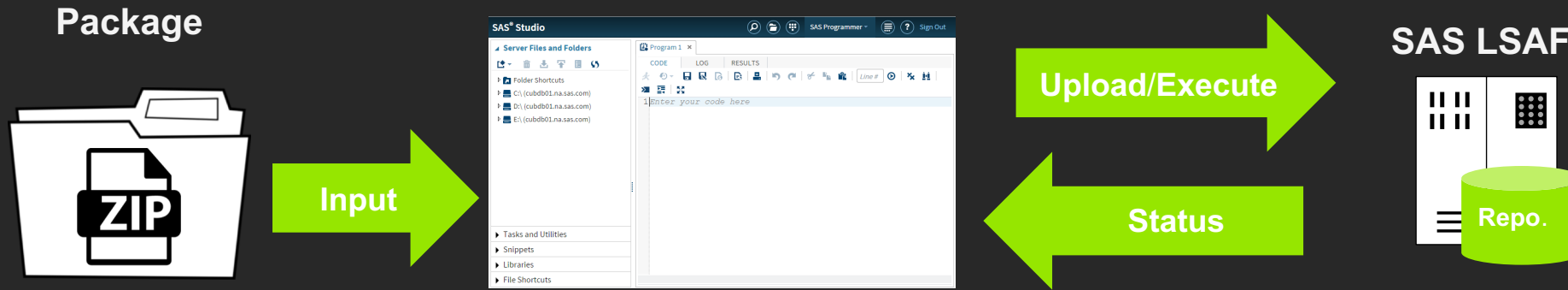


GRTCDW_CREATEPACKAGE:

- creates a ZIP file, which contains the required folder structure and all related files, which are listed in the job metadata file

```
1
2 /* Create package */
3 %grtc_dw_createpackage(mdspath=C:\dev\metadata,
4     mdds=t_ibu_tab07);
5
6
```

MACRO: TO EXECUTE A SAS LSAF JOB



GRTCDW_EXECUTEJOB:

- uploads the ZIP package to the SAS LSAF environment
- extracts the zip file and executes the job
- the macro creates also the needed folder structure
- creates a metadata QC file

EXAMPLE:

```
1
2 /* Upload package to the QC area and execute the job */
3 %grtc dw_executejob(cdwdir=/DUMMY/prg,
4     zippackage=prg\prg.zip,
5     cdwqcrun=1);
6
7
```

