



Job of Jobs

One job to rule them all

Kris Lauwers
Raphael Noirfalise

Jennifer Jacobs, *Stowaway*
Jennifer is a New York based artist
living with Type 1 diabetes.

Agenda

Introduction

Job of Jobs

Example 1: Clinical Study Report Preparation

- Running CDISC ADaM datasets & tables/listings/figures

Example 2: Running Simulations

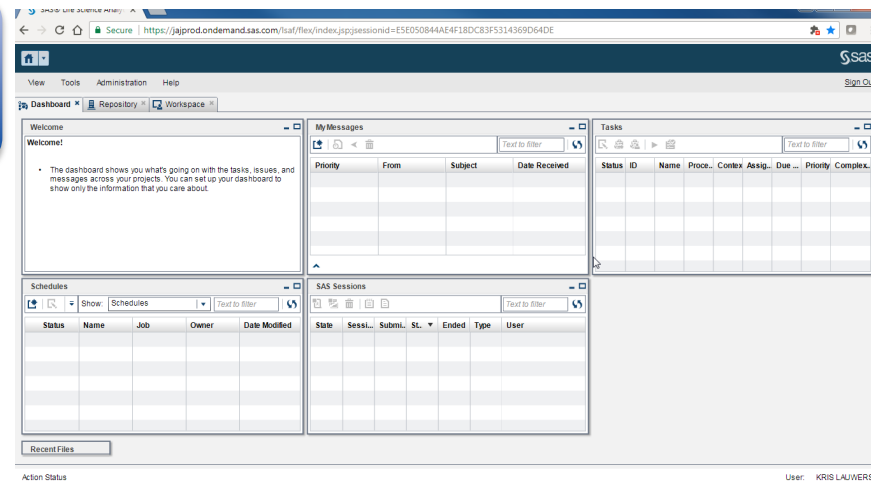
Conclusions

Introduction

Batch submit multiple SAS® programs

Platform: SAS Life Science Analytics Framework (LSAF)

- Creating a job:
 - Runs one or more SAS programs
 - Predefine global macro parameters
- Running a job:
 - LSAF will publish the output only when the job is complete.



Job of Jobs

One job that runs a SAS program in which other jobs are called

Makes use of LSAF Macro Application Programming Interface (API)

```
%lsaf_getchildren(  
  lsaf_path=,  
  sas_dsname=lsafGetChildren,  
  lsaf_recursive=0);
```

```
%lsaf_getjobparameters(  
  lsaf_path=,  
  lsaf_version=,  
  lsafGetJobParameters);
```

```
%lsaf_submitjob(  
  lsaf_path=,  
  lsaf_version=,  
  sas_dsname=sas_dsname=);
```

Allows for automatically submitting multiple SAS programs, either simultaneously or consecutively

Example 1: Clinical Study Report (CSR) Preparation

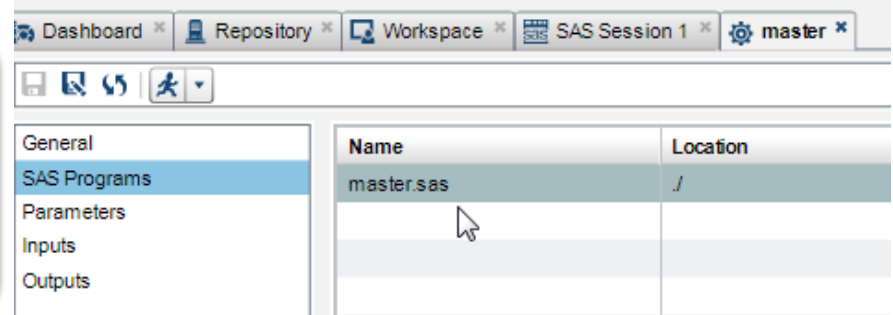
Creating CDISC ADaM datasets and all tables/listings/figures (TLF) needed for the Clinical Study Report after database lock.

- On critical path
- Running 100s of jobs and programs separately takes a lot of time and is error prone

Example 1: Clinical Study Report Preparation

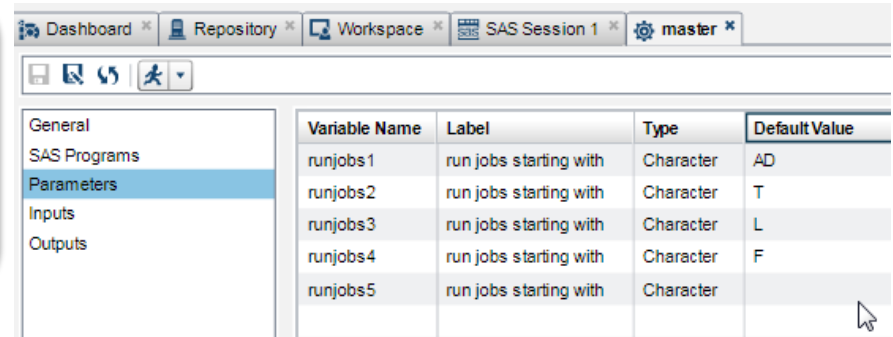
Create a job for each SAS program that generates one dataset or TLF and use consistent naming conventions (AD*, T*, L*, F*)

Create and run the parent job and program which in turn runs all the necessary datasets and TLFs for the CSR automatically



The screenshot shows the SAS Studio interface with the 'master.sas' job selected. The left sidebar shows the 'General' tab, and the right pane displays a table with the job details.

Name	Location
master.sas	./



The screenshot shows the SAS Studio interface with the 'Parameters' tab selected for the 'master.sas' job. The right pane displays a table with the job parameters.

Variable Name	Label	Type	Default Value
runjobs1	run jobs starting with	Character	AD
runjobs2	run jobs starting with	Character	T
runjobs3	run jobs starting with	Character	L
runjobs4	run jobs starting with	Character	F
runjobs5	run jobs starting with	Character	

Example 1: Clinical Study Report Preparation

```
*** master.sas;

*** get macro parameters from master.job;
%lsaf_getjobparameters(
  lsaf_path=%cmpres(&joblocation/master.job),
  sas_dsname=master_jobparameters);

data work.jobparameters;
  set master_jobparameters;
  rename defaultvalue=value;
run;

*** get all jobs related to each separate program;
*** you want to run; %lsaf_getchildren(lsaf_path=
%cmpres(&joblocation),
      lsaf_recursive=1,
      sas_dsname=jobs1);

*** select the jobs you want to run;
data jobs2;
  set jobs1;
  if name in:("%cmpres(&runjobs1)") or
    name in:("%cmpres(&runjobs2)") or
    name in:("%cmpres(&runjobs3)") or
    name in:("%cmpres(&runjobs4)") or
    name in:("%cmpres(&runjobs5)");
  if index(upcase(objecttype),"JOB")>0;
run;

*** count the number of jobs selected;
proc sql noprint;
  select count(*) into:njobs
  from jobs2;
```

```
*** run every job consecutively;
%macro loop;
  %do x=1 %to &njobs;

    data _null_;
      set jobs2;
      if _n_=&x then call symput("job&x",trim(left(name)));
    run;

    %lsaf_submitjob(
      lsaf_path=%cmpres(&joblocation/&&job&x),
      sas_dsname=work.jobparameters);

    *** put SAS to sleep until submission status of;
    *** &&job&x is set to "completed". Omitting this step;
    *** will let you run all selected jobs simultaneously;
    %tryAgain:

    *** let SAS sleep for 6 seconds;
    data _null_;
      call sleep(6,1);
    run;

    %lsaf_getsubmissionstatus(
      sdd_jobsubmission_id=&sddJobSubmissionId_);

    %if (%index(&_sddJobSubmissionStatus_,%STR(COMPL))<=0)
      %then %goto tryAgain;

  %end;
%mend loop;
```

Example 2: Running Simulations

Statisticians often need to run simulations with thousands of repeats

- On one local SAS machine: 100.000 simulations, 4 days (~100 hours) runtime, 16 simulations per minute
- On LSAF: output result files and logs can become huge while workspace on LSAF is limited

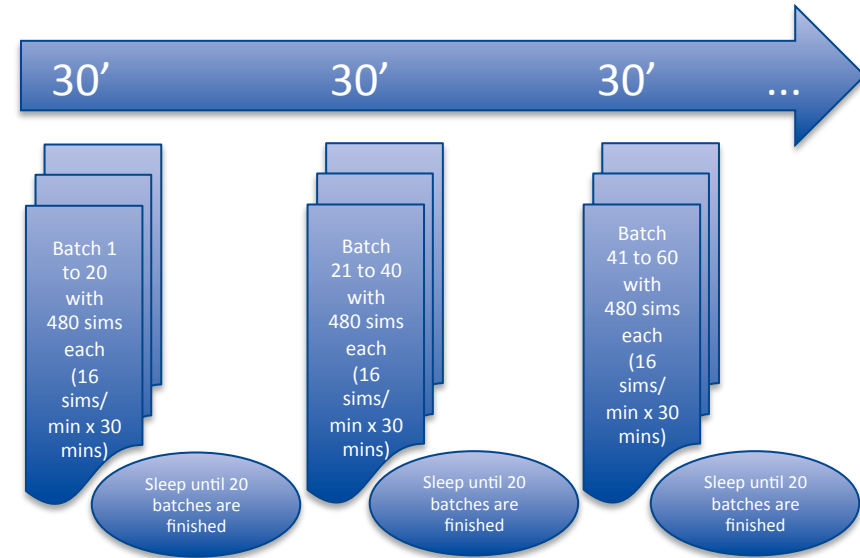
Example 2: Running Simulations

Employ multiple LSAF server cores at the same time by running jobs simultaneously

Outline of SAS program and job setup

- Break up simulations in batches running for 30 minutes and create a job for each batch
- Run 20 batch jobs simultaneously for 30 minutes
- Put SAS to sleep until all 20 batch jobs are successfully finished
- Run the 30 minutes schedule of 20 batch jobs consecutively until all simulations are done
- Compile all output datasets

Runtime is now 5 hours instead of 100 hours.



Conclusions

Easy to use tool that enables the users of cloud-based SAS environment to run multiple programs in batches

Applied throughout J&J to run CDISC ADaM dataset and TLF programs in one go after database lock.

Enables running large scale analysis or simulations



*Janssen is proud to feature artwork created by people
affected by the illnesses and diseases
we are committed to treating and preventing.*

Thank you

Kris Lauwers, Johnson and Johnson, Beerse, Belgium
klauwer2@its.jnj.com

October 10, 2017

janssen

PHARMACEUTICAL COMPANIES

OF *Johnson & Johnson*

