

BREAKING AWAY FROM SHELL SCRIPTS FOR BATCH MODE SUBMISSIONS ON SAS® GRID

US PhUSE Connect 2018

Paper TT12

Robert Diseker

June 3-6, 2018



Agenda

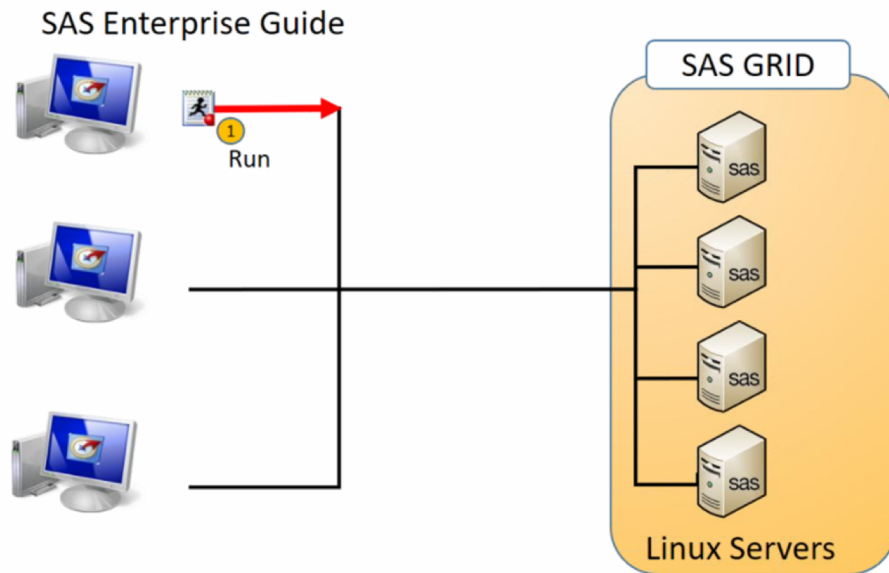
SAS GRID BATCH SUBMISSION MACROS

- ▶ Grid operation overview
- ▶ Processes needed to submit in batch mode on Grid
- ▶ Macros and their uses
- ▶ Scheduling jobs

SAS Grid Overview



How a SAS GRID works



SAS jobs are sent to any number of servers once programs are batch submitted, and they run at various speeds.

Definitions of Interactive vs. Batch Mode

► Interactive mode:

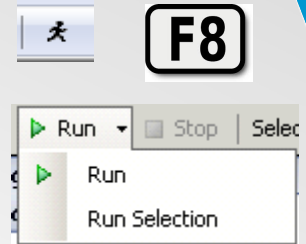
- Run/Submit a partial or complete program to the SAS Engine from an interactive session.
- Uses local SAS session WORK directory
- SAS engine returns Log and Results to interactive session

► Batch mode:

- Run/Submit a complete program to the server
- Uses server (not local) SAS session WORK directory
- Returns Log and Results as permanent files (*.log,*.lst)

► Batch of programs:

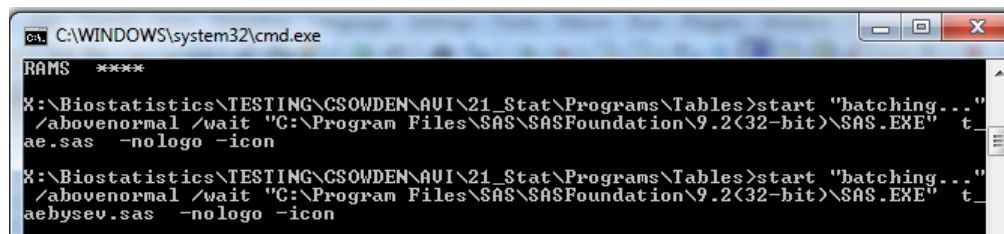
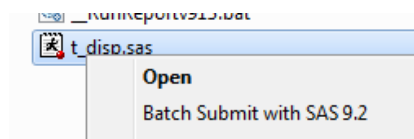
- Run/Submit a group of programs to the server
- In batch mode, a batch or shell script file customizes execution
 - In Windows that's a .bat file (double-click to execute)
 - In Unix/Linux that's a .sh or similar file (from command prompt)



Familiar Batch Mode Submission Processes

► Windows batch submission

- Right Click and batch submit a single program
- Use of .bat file containing all the files to be run in sequential order. Submitted by double-clicking in windows



```
C:\WINDOWS\system32\cmd.exe
RAMS ****
X:\Biostatistics\TESTING\CSOWDEN\AVI\21_Stat\Programs\Tables>start "batching..."
/abovenormal /wait "C:\Program Files\SAS\SASFoundation\9.2\32-bit\SAS.EXE" t_
ae.sas -nologo -icon
X:\Biostatistics\TESTING\CSOWDEN\AVI\21_Stat\Programs\Tables>start "batching..."
/abovenormal /wait "C:\Program Files\SAS\SASFoundation\9.2\32-bit\SAS.EXE" t_
aebysev.sas -nologo -icon
```

► Linux shell scripts run single/multiple programs from command prompt

```
#!/usr/bin/sh
dtstamp=$(date +%Y.%m.%d_%H.%M.%S)
pgmname="/sas/code/project1/program1.sas"
logname="/sas/code/project1/program1_`${dtstamp}.log"
/sas/SASHome/SASFoundation/9.4/sas $pgmname -log $lognam
```

Why run SAS programs in batch mode?

- ▶ Submitting multiple programs
 - Simultaneous execution to maximize power of the Grid
 - Sequential file execution
 - Order execution to account for data dependencies:
- ▶ Harness server computing power
- ▶ Documentation: Permanent SAS Log and Results files
- ▶ Automation: Scheduling

Moving from PC SAS to SAS Grid

PROCESSES NEEDED TO BATCH SUBMIT ON GRID

- ▶ Submit a single program in batch mode as a server job
- ▶ Monitor the progress of a server job
- ▶ Limit execution of SAS code until the server job completes.
- ▶ Create a “run-all” type program to submit batches of programs and control execution order

- ▶ Options available
 - Use shell scripts at the command prompt
 - SAS® Information Delivery Portal
 - SAS® Management Console-Batch scheduler
 - Load Sharing Facility (LSF) commands using EG macros to submit programs in batch mode from an interactive session

Platform Load Sharing Facility on SAS Grid

- ▶ The **Platform LSF** ("LSF", short for load sharing facility LSF) a workload management platform on networked unix, linux, and windows systems.
- ▶ LSF finds the best resources to run the job, and monitors its progress
- ▶ LSF commands, normally called from a command prompt, are accessed using a pipe command directly from SAS EG:
filename {dataset} pipe "LSF command";
- ▶ The macro suite makes use of two LSF commands in the interactive session:
 - **SASGSUB**: submits SAS jobs in batch mode to the servers
 - **BJOBS**, short for Batch Jobs: returns server job monitoring information

Submit a single program in batch mode as a server job

%GSUB({DIRECTORY PATH}/PROGRAM.SAS)

► SASGSUB Pipe Command

```
filename gsub pipe "&SASGSUB  
    -gridsubmitpgm    '&pgm'  
    -gridsasopts ""  
        -altlog '&plog'  
        -altprint '&plst'  
        -autoexec '&aepath' "" ";
```

► Macro Variables:

- SASGSUB -batch submission utility for SAS® version 9.4
- PLOG -path and name of log file
- PLST -path and name of lst file
- AEPATH -path and name of autoexec.sas

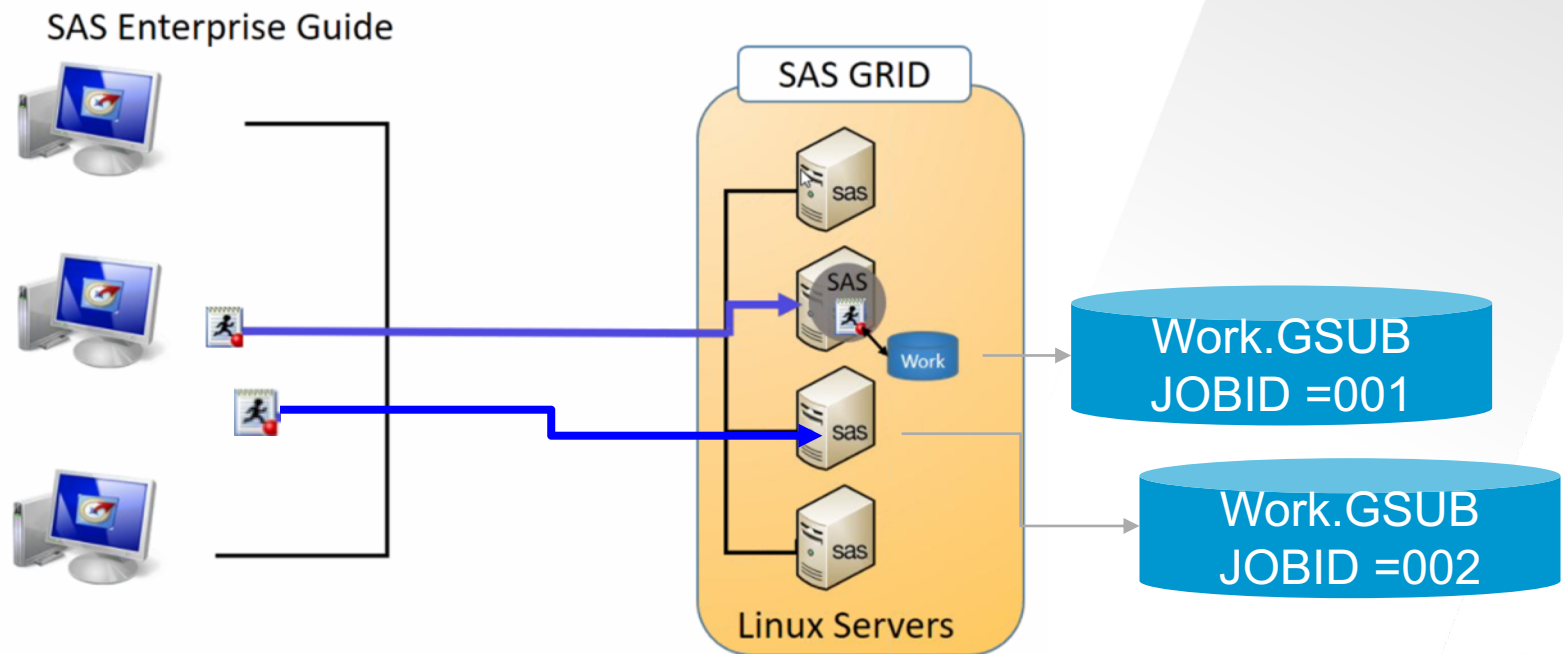
► No special code required in the SAS program being batched

%GSUB writes the server JOBID to the WORK.GSUB dataset

WORK.GSUB HOLDS A SINGLE JOBID

`%gsub(directory/adsl.sas)`

`%gsub(directory/adae.sas)`



Monitor the progress of a server job

BJOBS PIPE COMMAND

- ▶ The BJOBS command pipes information of the user's server jobs into a local JOBS dataset for future processing
- ▶ Returns a Status of RUN, DONE, EXIT, or PENDING for each job ID

```
filename jobs pipe "bjobs -a";  
data jobs;  
infile jobs firstobs=1 dlm=" " missover;  
length job_id $20. user $20. status $20.queue $20. sub_server $20.  
ex_server $20. jobname $20. month $20. day $20. time $20.;  
input job_id $ user $ status $ queue $ sub_server $ ex_server $  
jobname $ month $ day $ time $;  
run;
```

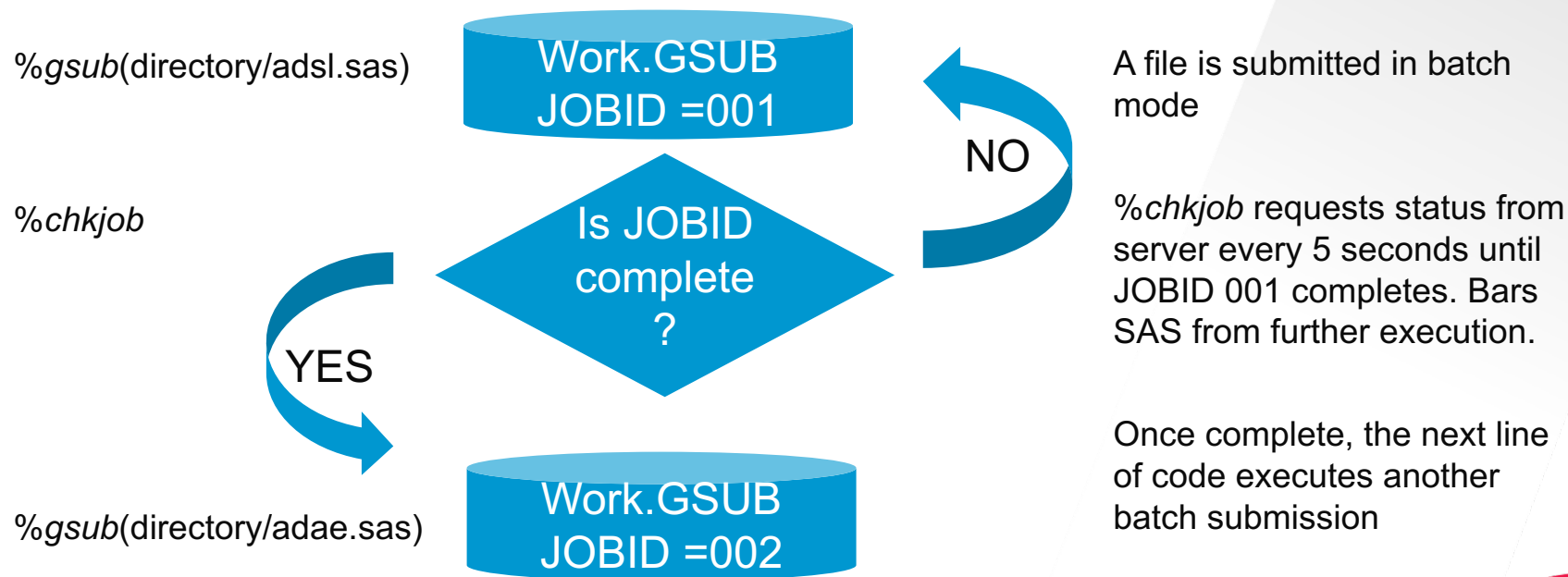
My Grid Jobs Today

Obs	job_id	user	status	queue	sub_server	ex_server	jobname	month	day	time
1	393403	radisek	RUN	normal	chprgc01	chprgn05	*2EE45C48F	May	15	14:52
2	394482	radisek	RUN	normal	chprgn05	chprgn08	table1	May	15	18:49
3	394484	radisek	DONE	normal	chprgn05	chprgn06	table3	May	15	18:49
4	394483	radisek	DONE	normal	chprgn05	chprgn03	table2	May	15	18:49
5	394486	radisek	DONE	normal	chprgn05	chprgn04	testtab	May	15	18:49
6	394485	radisek	DONE	normal	chprgn05	chprgn02	table4	May	15	18:49

Monitor the progress of a server job

%CHKJOB MACRO

- ▶ Tracks the progress of the most recent single file batch submission on the grid using the JOBID in WORK.GSUB merged with the BJOBS data
- ▶ Delays further SAS code from executing until the JOBID Status is DONE or EXIT or RUN



GROUPS OF PROGRAMS RUNNING SIMULTANEOUSLY

%APPENDGSUB AND %CHKALLJOBS MACROS

There is load balancing on the GRID to optimize throughput. To harness this power and still control program execution with batch submissions, we have 2 macros used together to ensure a group of batch jobs complete their execution on the grid.

- ▶ **%APPENDGSUB**: Create a list of batch jobs running on the Grid
- ▶ **%CHKALLJOBS**: Monitor these batch jobs and delay SAS line execution until they complete

%APPENDGSUB

CREATE A LIST OF BATCH JOBS RUNNING ON THE GRID

- ▶ From a succession of calls to `%gsub`, builds a list of submitted JOBIDs that are running on SAS® Grid concurrently.
- ▶ Appends each GSUB dataset to a new WORK.APPENDGSUB dataset
- ▶ Example: Run the production table programs before running the QC table programs

```
%gsub(&projroot./Programs/Tables/t_ae.sas)
```

```
%appendgsub;
```

```
%gsub(&projroot./Programs/Tables/t_cm.sas)
```

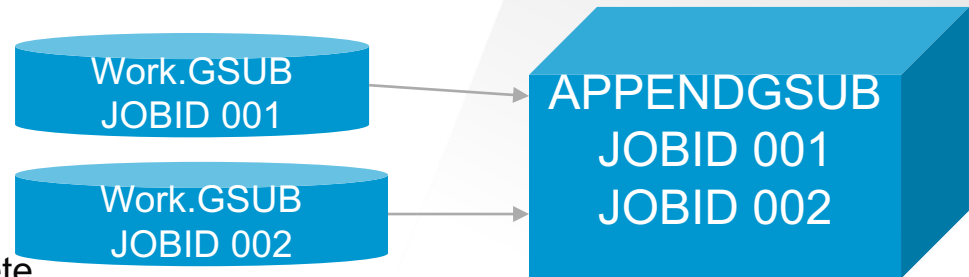
```
%appendgsub;
```

`%CHKALLJOBS` delays SAS line execution until both JOBIDs complete on SAS Grid

```
%chkalljobs
```

```
%gsub(&projroot./Programs/QC/Tables/vt_ae.sas)
```

```
%gsub(&projroot./Programs/QC/Tables/vt_cm.sas)
```



%CHKALLJOBS Functionality

- ▶ The macro checks the server until the number of pending jobs in the list from work.APPENDGSUB is 0.
- ▶ The next check to the server is delayed based on the number of pending jobs (review log note for progress).

Number of pending jobs	SAS Sleep time (seconds)
>100	50
>50	20
>20	10
>0	5

Note: [chkalljobs] loop #3, Time = 17:55:21, Elapsed time: 00:00:05 Number of pending jobs = 9

Note: [chkalljobs] Delay time to next server check = 5 seconds

- ▶ Once all pending jobs complete, the macro deletes work.APPENDGSUB and exits

%BRUNALL Creates a *%GSUB* call for all programs in a directory

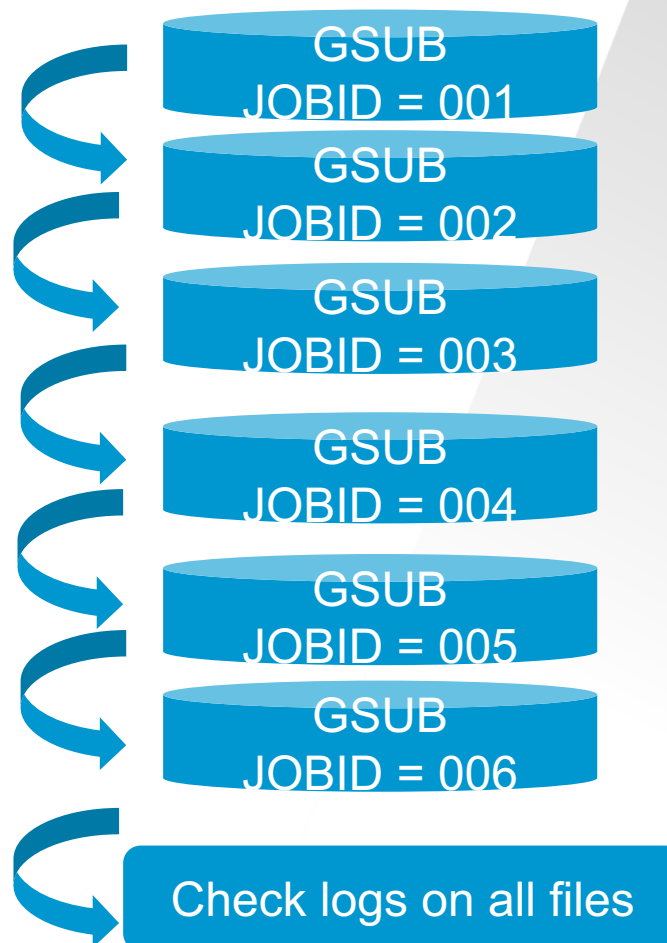
%BRUNALL MACRO

- ▶ Used for production runs of a directory of files
- ▶ Creates one SAS program of *%GSUB* calls for each program in a directory in alpha-numeric order
- ▶ Options to customize the type of run
 - **Simultaneous:** Adds *%APPENDGSUB* between *%GSUB* calls and *%CHECKALLJOBS* at the end
 - **Sequential:** Adds *%CHKJOB* between *%GSUB* calls
 - **Mixed:** Use a text file for customization of the run dependencies (mix of simultaneous and sequential)

Example of Sequential Batch File Execution for ADaM datasets

USE OF %CHKJOB

```
%gsub(&projroot./Programs/ADAM/01_ADSL.sas);  
%chkjob;  
%gsub(&projroot./Programs/ADAM/02_ADCM.sas);  
%chkjob;  
%gsub(&projroot./Programs/ADAM/ADAE.sas);  
%chkjob;  
%gsub(&projroot./Programs/ADAM/ADAUD.sas);  
%chkjob;  
%gsub(&projroot./Programs/ADAM/ADLB.sas);  
%chkjob;  
%gsub(&projroot./Programs/ADAM/ADPE.sas);  
%chkjob;  
%gsub(&projroot./Programs/ADAM/ADQS.sas);  
%chkjob;  
  
**** Log Check File ****;  
%logchk(&projroot./Programs/ADAM);
```



Example of Multiple Batch File Execution of tables running simultaneously

USE OF %APPENDGSUB AND &CHKALLJOBS

```
%gsub(&projroot./Programs/Tables/t_ae.sas);  
%appendgsub;
```

```
%gsub(&projroot./Programs/Tables/t_ae_cyc.sas);  
%appendgsub;
```

```
%gsub(&projroot./Programs/Tables/t_ae_ov.sas);  
%appendgsub;
```

```
%gsub(&projroot./Programs/Tables/t_ae_ov_cyc.sas);  
%appendgsub;
```

```
%gsub(&projroot./Programs/Tables/t_ae_rel.sas);  
%appendgsub;
```

```
%gsub(&projroot./Programs/Tables/t_ae_sev.sas);  
%appendgsub;
```

```
%chkalljobs;
```

```
**** Log Check File ****;  
%logchk(&projroot./Programs/Tables);
```

Batch submit
all table files
at once

WORK.APPENDGSUB

JOBID

001

002

003

004

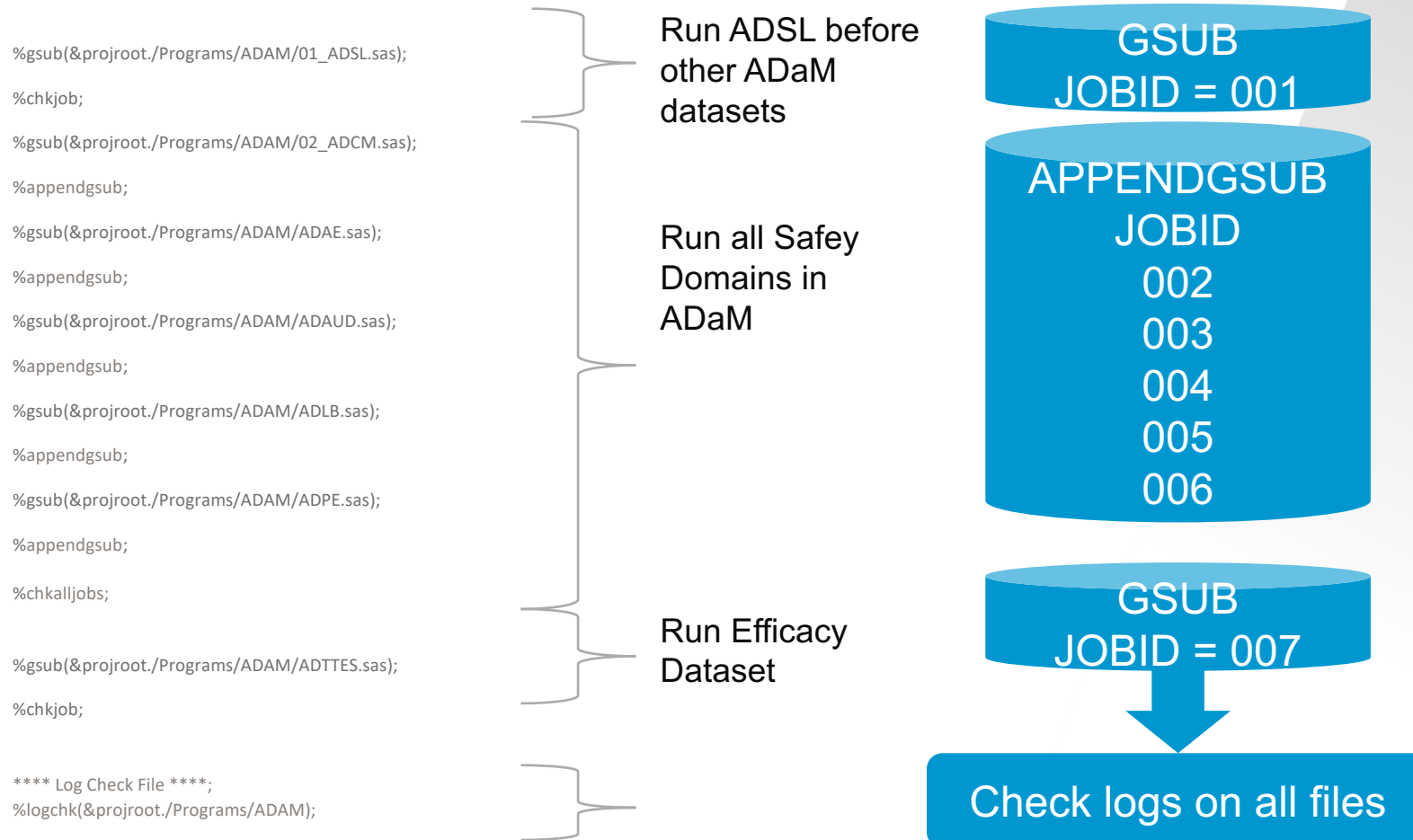
005

006

Delay execution until all JOBIDS
complete in the grid

Check logs on all files

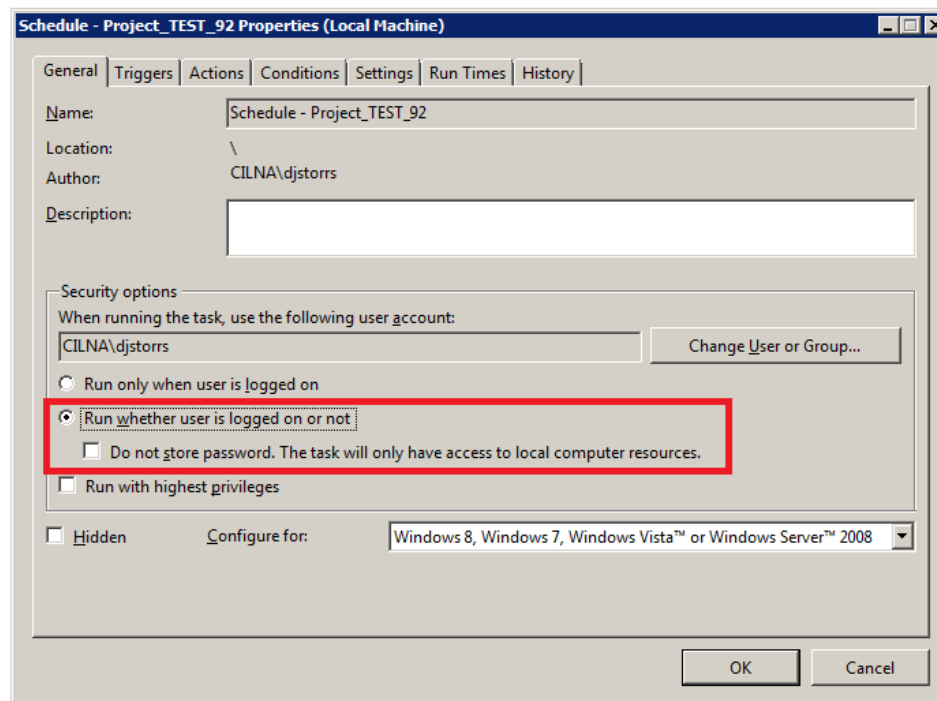
Example of Customized Batch Execution for ADaM datasets with dependencies



Schedule jobs using EG Scheduler

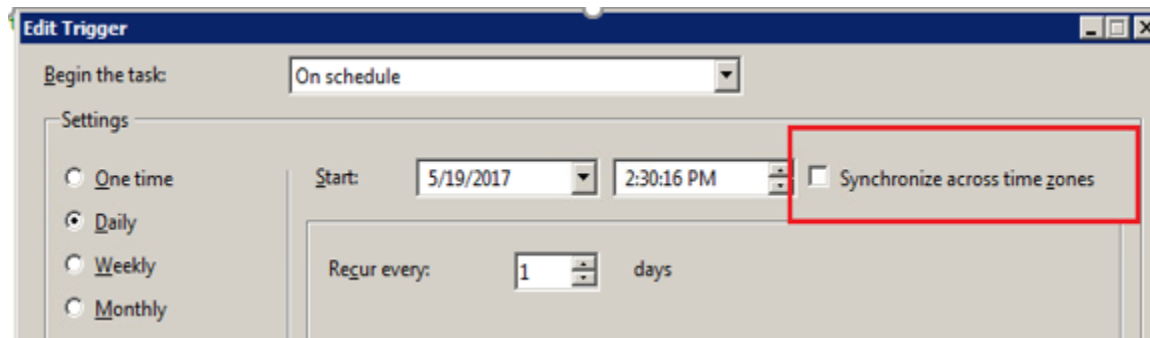
NOTE: IT SETUP MAY BE REQUIRED

- ▶ Create an EG Project and save it
- ▶ Use the built in scheduler: EG File> Schedule
- ▶ Check the radio button “Run whether User is logged on or not.”



Schedule jobs using EG Scheduler

- ▶ Edit the Trigger tab to set the time and turn off the Synchronize across time zones



- ▶ Select Properties and click on the Embed button (it will be grayed out after selected). Then click OK.
- ▶ When the Schedules window is closed, the user will be prompted for credentials.
- ▶ Save Project (File → Save all) then log completely out of EG
- ▶ The File will run as scheduled

Summary: SAS Batch Submission Macro Suite

- ▶ Macros designed to customize batch execution
- ▶ Quickly create a “Run-all” type program to fit the programmers needs for production runs
- ▶ Schedule batch submission using EG Scheduler
- ▶ Macro suite is ideal for SAS Grid systems
 - Avoid shell scripts on the command line
 - Avoid IT, setup delays and permissions difficulties of SAS Management Console and SAS® Information Delivery Portal
 - No special code required in individual programs
 - Enterprise Guide interface for interactive programming

Questions?



Covance Inc. and Chiltern-A Covance Company are the global drug development businesses of Laboratory Corporation of America Holdings (LabCorp). COVANCE, CHILTERN and DESIGNED AROUND YOU are registered trademarks and the marketing names used by Covance Inc. and its subsidiaries around the world.

THE AMERICAS +1.888.COVANCE +1.609.452.4440 or +1.910.338.4760

EUROPE/AFRICA +00.800.2682.2682 +44.1423.500888 or +44.1753.512000

ASIA PACIFIC +800.6568.3000 +65.6.5686588