SOP 099 – Usage Korn .KSH Linux Scripts and Toolbox

1. **Purpose**:

This documentation provides an overview of the usage of Korn shell scripts to automate SAEB processing on the SAE server and the toolbox script repository to automate various SAS functions and to standardize log checks.

1. **When**:

KSH scripts are useful for running several related SAS programs from start to finish. This ensures that the set of programs are run in the proper sequence. This is the preferable method for ALL SAEB production programs.

1. **Who:**

|  |  |
| --- | --- |
| **Level** | **Staff – Role** |
| Familiar | SAEB General Staff  SALE ADC |
| Working | SAEB Branch Chief  ALL SAEB Staff that run production programs |
| Expert | SAEB Dissemination SME/Team Lead  SAEB SAHIE/SAIPE Production Team Lead(s) |

1. **Scope of Work**:

This SOP covers ALL SAEB production programs (SAS, R, and C)

The usage of a driver .ksh files **is suggested** but not mandatory.

The usage of the do\_one\_sas\_batch.ksh script for SAS programs **is required**.

1. **What: (Procedure)**

All SAEB staff are responsible for the best set of ‘problematic’ words. But since the do\_one\_sas\_batch.ksh file is under configuration management (CM), an SAEB Issue must be initiated to add/subtract target phrases and/or change their severity level.

Access to the toolbox from any terminal is provided by editing the user’s .bashrc file **once** by running the following command:

echo 'export PATH=$PATH:/sae/admin/toolbox' >> ~/.bashrc

1. **An Example Korn Shell (.ksh) Script**

Unix/Linux had several scripting languages available to bundle several commands into a single command (Windows only has 1: DOS). For no particular reason, we decided to use Korn Shell (.ksh) scripts.

#!/bin/ksh

# 00\_sahie\_dissemination\_singleyear\_driver.ksh

#

# to run type ./{progname}.ksh (after making priv executable)

do\_one\_sas\_batch.ksh 01\_making\_sahie\_singleyear

do\_one\_sas\_batch.ksh 02\_qc\_sahie\_singleyear

do\_one\_sas\_batch.ksh 03\_write\_sahie\_singleyear\_txt

do\_one\_sas\_batch.ksh 04\_write\_sahie\_singleyear\_csv

check\_all\_logs.ksh

The above example Korn Shell script runs the 4 SAS program in sequence that perform the SAHIE Single Year processing.

Some details to note:

* The first line “#!/bin/ksh” is required – it indicates that the file is a Korn Shell Script.
* Comments are created using the “#” symbol (except as seen on the first line).
* do\_one\_sas\_batch.ksh is a SAEB custom written Korn Shell program that runs the program but then checks the .log for about 15 problematic phrases.
* check\_all\_logs.ksh checks the log files one more time and saves the output for record keeping.
* A .ksh file must have ‘eXecution’ privileges to run. See chmod command documentation.

1. **SAS\_LOG\_CHECK & DO\_ONE\_SAS\_BATCH Korn Shell (.ksh) Script**

This Korn shell script can either be run from another .ksh script or run directly from the command line. Its main advantage is that it automatically checks the .log file for known SAS problems – EACH TIME.

SAEB worked with other branches to implement what we felt is the best set of things to check. This script is called by various other “flavors” of SAS scripts so that only one version needs to be maintained. The internal workings of the script are out of scope for this document but the following are the items currently (July 2015) checked:

**SEVERE Checks - Must be corrected** – if not possible **Branch Chief Approval required**.

ERROR: an error, typically means the program did not complete

warning common sign of a critical problem

uninit For Uninitialized

truncate Especially when reading a .TXT or CSV file (ASCII – Flat File)

repeats of BY values Merge probably giving incorrect results

**HIGHLY Problematic Checks** – Should be corrected - Should be ‘rare’.

if not possible note and get peer reviewer approval.

Invalid argument

End of line

does not exist

Lost card

Missing values

Multiple lengths

Not resolved

Invalid numeric data

SAEB NOTE: Custom flag that will show up in log checker, must be written into program by SAE

**LOW Items Checked**

**Allowable but should be researched as to the cause and validated by peer reviewer.**

converted

Character values

No observations This might be okay – dataset of bad records should empty.

0 observations Same as No observations

“Does not exist” during a Proc Append Example:

# NOTE: Appending WORK.C\_RECODE\_POV\_SE to WORK.C\_APPEND\_POV\_SE.

# NOTE: BASE data set does not exist. DATA file is being copied to BASE file.

1. **Other toolbox scripts:**

Available scripts and a short description of each, see the script header for details:

check\_all\_logs.ksh - runs the log\_check for every SAS program in the current directory and saves output to a permanent file

do\_one\_sas\_batch.ksh - used to execute the SAS program in batch mode and then check the log for any problems, SAS command line options can be passed after the program name

example\_driver.ksh - not a functional script but an example of a script that runs multiple commands in a specific order

run\_sas\_with\_memsize.ksh - executes SAS with memsize set to MAX unless user specifies the size

run\_sas\_with\_r.ksh - executes SAS using the "rlang" option so that R code can be used

sae\_permissions.ksh - secures all files within current or specified directory

sas\_log\_check.ksh - searches for various messages in the SAS log that indicate a problem