

Session 1634: Productionalizing SAS® for Enterprise Efficiency at Kaiser Permanente

Ryan Henderson
Karl Petith

Potential
of One

Power
of
AI



Productionalizing SAS® for Enterprise Efficiency

Ryan Henderson, Karl Petith

Kaiser Permanente

Abstract

•Kaiser Permanente Northwest (KPNW) has designed standardized processes and procedures that have allowed KPNW to streamline the support of production SAS® content, which enabled KPNW analytical resources to focus more on new content development rather than on maintenance and support of steady state programs and processes. We started with over 200 individual SAS processes across four different SAS platforms, SAS® Enterprise Guide, Mainframe SAS®, PC SAS® and SAS® Data Integration Studio, in order to standardize our development approach on SAS® Enterprise Guide and build efficient and scalable processes within our department and across the region.

Objective

Leverage a consistent, repeatable and documented production support process that enables DIME to be proactive in content management and business partner interaction to achieve confidence and reliability in all DIME content.

- Standard Repeatable Process
- Scheduled and Automated
- Monitor by Exception
- Maintainable
- Measurable
- Scalable

Results

- 130+ jobs running in Production Support
- 800+ completed processes a month
- 98%+ First run success rate
- Documented dependencies allow for impact analysis quickly and easily
- Support SAS processes and reports that are developed outside our department



Productionalizing SAS® for Enterprise Efficiency

Ryan Henderson, Karl Petith

Kaiser Permanente

Method

- **Standard Job Header**

Production jobs have a standard job header. This is a small summary that is inserted into every job, so that some key pieces of information are collected. The header has enough information for Production Team members to provide Tier 1 support without having detailed knowledge about the job.

Sample Job Header

Job: nwdime_ac_999_primary_care_department_list

Description: Sample job that produces a list of primary care departments.
Mailed to business partners and put on the shared drive with FTP.

Schedule: 1st of every month

Recovery: Rerun job

Dependencies: Depends on daily update of Clarity data.

Cognos usage: Package:
Cube:
Reports:

Source Tables: CLARITY
hcclnw.clarity_dep

Output: REPORT
clarity_pc_departments_YYYY_MM_DD.csv report (emailed)

Created: 4/19/2013

Created By: Karl Petith

DIME Owner: Karl Petith

Business Partner: Ryan Henderson

Modifications: 5/28/13 Karl Petith - Revised sample code.

- **Standard Login Macro Variables**

Production jobs use standardized login, password, database, and schema macro variables. Passwords should not be present in SAS code. Logins/passwords/schemas are used in jobs as macros that reference the group login file or individual login file. The Production team can provide a standardized individual login file that can be placed in a developer's personal folder on Unix and modified to use the developer's personal user id, passwords, etc.. The login files are accessed using an %INCLUDE statement in the code. If Libname statements and Connect To statements (for SQL Pass Through) use these standard macros, then they can be easily transitioned between Production and Development teams.

Before jobs are scheduled, they should be tested with the group logins. All jobs in Production run under group logins.

Productionalizing SAS® for Enterprise Efficiency

Ryan Henderson, Karl Petith

Kaiser Permanente

Method cont'd

- Output Naming

Output files should be named so that they can be easily archived and sort correctly. Often a job that produces a report will just overwrite the file from a previous run; If possible, the Production team is trying to archive report output for files that are not too large.

The standard format is report_name_YYYY-MM-DD (this is SAS yymmddd10. format), but this is just a suggestion. Output files can include uppercase characters and spaces in their names.

Output naming example:

```
proc export data=INF_PREV2
      outfile="&outputpath/Surgical Data &rundate..csv"
      dbms=csv replace;
run;
```

- Long Term Data Storage

While it is often convenient for developers working on code to store SAS datasets and tables locally (e.g. *.SAS7* files), production jobs code should try to use work tables only. If data needs to be stored, storage in a database under a group schema (e.g. NWDW) is preferred over local storage in the Unix environment.

- Log Files

Log file output is directed to a specific folder by preprocess code after a job is moved to production. The preprocess code is added when a job is deployed, so there is no need for the developer to worry about where the logs are directed. The log files are all archived, so if there is a problem both the most recent and historical logs can be given to a developer to review.

If an error is detected in a production process, the log file is automatically sent to the Production Support team for analysis and recovery.



Productionalizing SAS® for Enterprise Efficiency

Ryan Henderson, Karl Petith

Kaiser Permanente

Method cont'd

- **Automation Standards**

- Jobs must run without errors.
 - It is helpful to also eliminate warnings from code, because warnings make monitoring more difficult.
- Code should be standardized.
 - Logins, paths, date ranges are all defined at the start of code
 - Standard Libname or Connect To (SQL Pass Through) statements used for database connections
- Jobs need to be able to run on a schedule with no user input.
 - For legacy jobs that have a manual component, the Production team can review and suggest automation options given the tools that are available.
 - Date ranges should be relative, so they do not need to be changed each run of the job. Date start/stop definitions should be called at the start of the job (like the paths or login file). Many relative date ranges can be defined with the very useful INTNX function.
 - So far, several manual jobs have been automated by changing output formats, putting in conditional logic to check for table updates, and general code cleanup & consolidation.
- Job output should not require any additional manual work by production team.
 - There must be no manipulation of output files, including running Excel macros, copy/pasting, moving files, formatting, etc..
- Output distribution and archiving will be automated.
 - Email (with attachments), archiving of output files, FTP file transfers, dropping files on shared drives can be automated.



Washington, D.C.

March 23–26, 2014