

SAS Programming Standards

Structure

- Apps are coded so that each “logical step” is coded as a physically separate SAS program. Each logical step is stored as a member of a collective in a SAS catalog or operating system folder. Each catalog or folder is easily mapped to a specific App and/or its categorical function across Apps.
 - Each logical step is programmed to evaluate a global AppCC that indicates error and normal operating condition.
 - When executed, each logical step is programmed to set and evaluate a local programCC that indicates error or normal operating condition.
 - Each logical step is programmed evaluate a local enable/disable flag that indicates “execute normally” or “do not execute”, which may be set/reset for each observation.
- Each logical step is identified in the Program Work List.
 - Contains the name, physical location, and sequence of execution of every SAS program in the App.
 - Logical steps do NOT contain import or include statements.
 - Logical steps do NOT dynamically generate code.
 - Logical steps that are object oriented do NOT perform instantiation of other classes.
- Prohibit compiled macros and include statements in new code.
 - Refactor the calling program and %include into a series of standalone programs executed in sequence through the Program Work List. Nested references amplify “spaghetti code” effect and plainly indicate where the Apps need to be refactored.
- Local variables that control execution of methods/steps within each logical step are explicitly PUT to the SAS Log.
 - Variables are not renamed when used as keyword or positional parameters on a macro invocation to ensure easy debug/tracing
 - Messages and unique ID’s indicate meaning for values in local control variables that can be interpreted by QA during defect analysis.
- No SAS program creates error or warning messages that are considered “normal”. The program contains additional messages to indicate that condition is acceptable, when QA teams must take action, what action is necessary.
- There is no “commented code” - anywhere.

SAS Application Controller

- Reads the Program Work List and executes each SAS program in the required sequence.
 - Runs a standardized routine to capture errors, warnings, and QA message(s) from each logical step.
- Format the SAS Log for easy reading
 - Add lines to indicate Program Start, Program End, complete program name, and physical location.
- Creates a SAS transport file on the following conditions
 - When any Data Step or Proc crashes.
 - When the program throws a specifically formatted error message.
- Global variables created by any program in the App are identified and PUT to the SAS Log after each program in the Program Work List is executed.
- Terminates the App immediately on the following conditions to eliminate logging and cascade error messages.
 - After any Data Step or Proc crash.
 - After any program throws a specifically formatted error message or sets the AppCC control variable.

General Requirements

- Each program self-disables on the following conditions.
 - Any prior program failed or set the global AppCC (global control).
 - Any prior step failed or set local programCC (local control).
 - The local control flag indicates “no need to execute”.
 - Required global variable(s) do not exist.
 - Required SAS dataset(s) do not exist.
 - Required SAS dataset column(s) do not exist.
 - Required folders do not exist.
- Each program provides LOG messages on for QA and defect analysis.
 - Messages are created for capture by the App Main Driver.
 - Messages indicate why the program failed or self-stopped.
 - Messages indicate program name, method name, or step within the program.
 - Messages include external RDMBS messages.
 - Messages indicate when the program did NOT execute, but the condition is NORMAL and expected.
 - Messages identify NOTES and ALERTs that can be used by QA for defect analysis.
- Each Data Step and Proc can be verified for the following.
 - Verify datasets contain data.

- Verify datasets contains the correct number of rows.
 - Verify each data step and procedure block is syntax correct.
- Create a unique error message that indicates program name, method name, step, or line number following every syntax and error check.

SAS Log

- Every component in the Program Work List is listed before App execution starts.
- Log contains lines that clearly indicate SAS program name, start point, and stop point indicator, for each logical step.
- Log contains a complete list of global variables after each SAS program is executed.
- Logging stops immediately when a syntax error is detected.
- Logging can be stopped immediately when a validation error is detected.

Development and Defect Testing

- The controller automatically detects when the App is running in Test Mode.
- Global variables, parameters, and datasets are created so the programmer can test and fix defects without altering the existing SAS code.
- Any SAS program in the Program Work List can be executed and tested for expected results without running the entire App.
- Any SAS program in the Program Work List can be updated without recompiling or rebuilding the entire App.

Integration

- The SAS App can be independently tested for expected results.