Introduction
When starting to use a new system, tool or software, certain qualification checks ought to be performed to ensure its correct work, such as design qualification (DQ), installation qualification (IQ), operational qualification (OQ), process qualification (PQ) and performance qualification (P1Q). SAS provides sufficient tools for performing most of these checks, such as SAS Installation Qualification Tool (SAS IQ) and the SAS Operational Qualification Tool (SAS OQ). These tools support the qualification aspect of the essential migration, integration, and verification processes users need to move from previous versions of SAS to later releases. However, performance qualification cannot be unified in the same way. It implies field-specified tests that check the system performance in real-world scenarios. This poster presents a program code designed to run performance qualification tests that have been selected taking into account the specifics of clinical programming. Methods used to perform these tests are described, as well as the produced results. The presented program code cannot only be used for performance qualification checks, but it presents an opportunity to compare the efficiency of different programming approaches and SAS procedures, which can be useful in everyday work.

Objective
The program code performs qualification tests that check calculations, procedures, methods, etc., that are used the most by a programmer in order to create statistical outputs for Clinical Study Reports (CSRs).

Methods
1. Initializing the logging facility for SAS programs is necessary if you use the logging facility autcall macros. The %LOG4SAS macro initializes the logging environment, and the %LOG4SAS_LOGGER macro defines a logger. We use the option “level = info” in order to get information that highlights the progress of an application.
2. ARM (Application Response Measurement) is an application programming interface which is used to monitor the availability and performance of software applications. ARM allows to measure application availability, performance, usage, and transaction response time. The %PERFINIT macro names the application instance and initializes the ARM interface.
3. The %PERFSTRT macro specifies the start of a transaction. Within each transaction, one of the programming tasks is performed. The %PERFSTOP macro specifies the end of a transaction.
4. The %PERFEND macro indicates the termination of the application. The %PERFEND macro means that the application does not issue any more ARM calls.
5. The %ARMPROC macro processes the ARM log and creates six SAS data sets that contain the information from the log.
6. The %ARMJOIN macro reads the six SAS data sets that are created by the %ARMPROC macro. It merges the information from those data sets to create data sets and SAS views for easier reporting of ARM data (SQL views).

Tasks
The following tasks have been identified as the most common and which performance is crucial for a clinical programmer’s routine work:
1. Creating a table on the picture below.
2. Performing descriptive statistics using proc means;
3. Calculating frequency statistics using proc freq;
4. Performing t tests using proc ttest;
5. Performing nonparametric tests using proc npar1way;
6. Performing ANCOVA with random factor using proc mixed;
7. Creating a table using proc report;

ARM Log
While each performance check is carried out, the result is written into the log.

Metrics
- JOCOUNT_ COUNT64: The total number of disk, tape, or related input and output operations at each %PERFSTRT and %PERFSTOP event
- MEMCURR_ GAUGE64: The current value for memory used in the process at each %PERFSTRT and %PERFSTOP event
- MEMHIGH_ GAUGE64: The highest amount of memory used for the life cycle of the current process at each ARM event
- THREADCURR_ GAUGE32: The current thread count of the process at each ARM event
- THREADHIGH_ GAUGE32: The highest number of active threads for the life cycle of the current process at each ARM event

Possible uses
This technique is useful in all tasks related to measuring the performance of any SAS codes. Except the main task (validation of the SAS platform or PQ check documentation) it can be successfully used in:
- Checking the SAS performance on different devices;
- Comparing different approaches of problem solving;
- Choosing the most effective algorithm for each programming task.

In case you need to produce PQ documentation, assistance in writing your code or in interpreting the results, feel free to send your request to us via email.

Results
The %ARMPROC and %ARMJOIN macros produce the results that are usually used for writing PQ documentation. Results are attached as appendices to PQ documentation.

Authors: Tatiana Volodina volodina.tatyana@gmail.com, Andrey Myslivets andrewmyslivets@gmail.com, Data MATRIX Ltd.