Paper 124-2009

Building a Data Repository Using Base SAS®

Ellen Krawiec, Monitoring Analytics, LLC, Eagleville, PA

ABSTRACT

Monitoring Analytics, LLC, (MA), provides independent market monitoring services to PJM Interconnection, L.L.C. MA uses PJM electricity market data drawn from nearly all of PJM's data stores and other external data sources. Rather than have MMU analysts query production databases directly, the MMU implemented and maintains the Market Monitoring System, (MMS), an Oracle database repository populated using Base SAS® and various SAS/ACCESS® products. The MMS copies data daily from PJM databases in original form to the MMS database. Metrics based on the source data copies and other metrics are calculated in the MMS SAS environment and stored in the MMS database.

The MMS consists of a custom, robust infrastructure, including the following:

- Framework that easily allows integration of new SAS programs
- Standardized SAS macro library to perform:

  Database updates, including data versioning;
  Timezone conversion;
  Error checking; and
  Email notification of job status.

- Table-driven job scheduling and workflow including:

  Program priority;
  Predecessor/successor relationships among programs;
  Control over the maximum number of concurrent MMS programs and
  Auditing of job completion status.

INTRODUCTION

Base SAS® and various SAS/ACCESS® products were selected five years ago because of the volume of data processed each day, and the complexity of the data transformations, and continue to be used to implement the Market Monitoring System, (MMS). The MMS consists of a SAS code framework which utilizes a SAS macro library and approximately 600 MMS programs. The execution of the programs is governed by configuration information stored in infrastructure tables stored in an Oracle database. The programs may run daily, monthly or annually to populate data tables stored in an Oracle database which currently consumes eleven terabytes of disk storage.

Several of the MMS programs copy data from tables in PJM's production databases to corresponding tables in the MMS database. The programs are called atomic, or base-level, programs. Other programs use data in the MMS database to perform calculations and transformations and store the results to the MMS database. The programs generate metric data. The MMS architecture is shown in Figure 1.

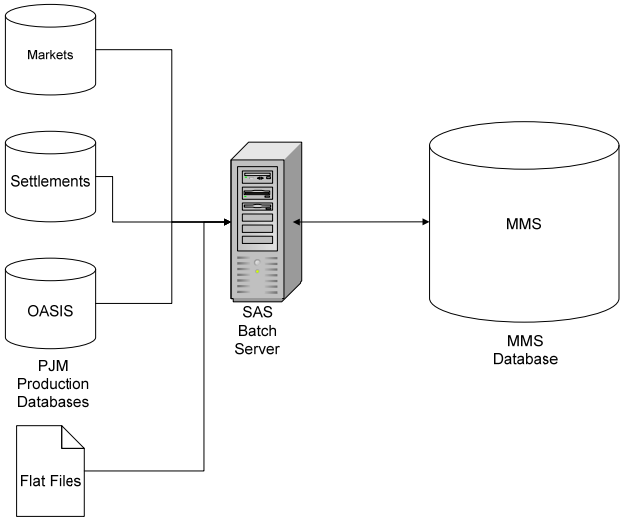

Figure 1: MMS Architecture

MMS FRAMEWORK

At the most basic level, the MMS is comprised of three major types of components, the scheduler, the kickoff program and the transformation programs, as shown in Figure 2. All of the components run on the SAS batch server.

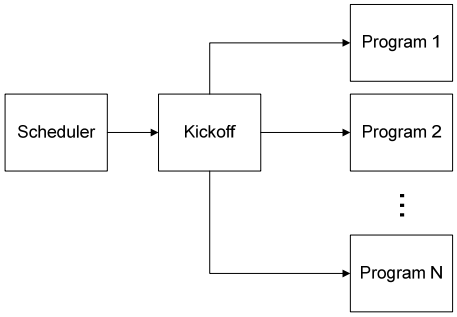

Figure 2: MMS Components

The scheduler, which is third party software, initiates the kickoff program at defined intervals. The kickoff program selects jobs from a queue table, and runs up to a specified number of transformation programs each interval.

Each time the kickoff program runs, it retrieves the following configuration information from the database to determine how many and which programs it will run:

- Maximum number of jobs that can run at the current time. The MMS accommodates a different value for each hour of each day. This is the MMS "throttle," allowing more programs to start in off-hours.
- The list of jobs in the queue table that are ready run. Using the program specific parameters, the central job schedule table is populated in advance. This table contains job entries for all of the programs that run each day, the program's earliest start time, the date range for the program to process, and the status of the job.
- Program specific information specific to starting each program:

  Program location: The Windows directory location is stored.
  Priority: An integer value ranging from 1 to 10, with 1 being the highest priority.
  Status: Set to either ACTIVE or INACTIVE. The kickoff program will select only ACTIVE programs.
  Earliest start time: The kickoff program will select only programs whose earliest start time has passed.

- Workflow information: Metadata that describes the order in which programs execute.

After the kickoff program selects the programs to run, it sets each job status to indicate that it has been selected, and initiates each program.

Each MMS program updates one database table, and is structured the same way and performs the same basic set of steps:

- Record program start time in the audit tables.
- Retrieve program-specific parameter:

  Date range used by the program,
  Target database and schema
  Email addresses for notification

- Retrieve common and program-specific constants from the database. Examples are locations of flat files that will be parsed and loaded.
- Connect to source database(s), source flat file directories and target databases.
- Check source table metadata: The MMS maintains metadata about all source database tables from which it retrieves data. If the table structure of the source tables changes, email notification is sent to the MMS operator.
- Retrieve data from the source tables.
- Perform program-specific transformations and calculations and prepare results for database.
- Update the database if the resulting dataset is different from what is currently stored in the database.
- Change the status in the queue table, and record row counts, errors and program end time in the audit tables.

METHODOLOGY

The MMS follows a strict methodology for code development and deployment. Any SAS program or macro must go through the following steps:

- Code design documented in flowcharts and process flow diagrams.
- Code implementation in the development environment.
- Preparation of a test plan.
- Review of code and test plan.
- Unit testing in the development environment.
- Migration to the test environment.
- Integration testing in the test environment.
- Migration to the production environment.

MMS MACRO LIBRARY

Since each program is very similar, a SAS utility macro library is utilized extensively in the MMS.

DATABASE UPDATES

There are two general types of data stored in the MMS database, interval data and slowly changing dimensional data.

- Interval data. At PJM, common intervals are five-minute, hourly, daily, and monthly. Tables containing interval data will usually have entries for each interval. An example of interval data is pricing data for which the database table will store a row for each price for each hour. The MMS goal is to maintain a copy of the production system. If the production system changes, the MMS replaces the stale rows with the updated data, (the stale rows are not versioned).
- Slowly-changing dimensional data. This data may also be referred to as “code” or “lookup” data. It may not change often, and usually has effective and termination dates. An example of dimensional data is company information like name and address. When the MMS copies this data, the macro code attaches MMS-specified effective and termination dates, thereby versioning the data.

Both types of updates are performed by SAS macros. The macros compare the contents of the source database and what is in the MMS using PROC COMPARE. There are different versions of each macro depending on whether the data is uniquely identified by one column or a combination of columns. Database updates occur only if the data in the two sets are different.

TIMEZONE CONVERSION

PJM does not enforce a standard time convention for its production systems; some are in GMT and some are in Eastern prevailing time, with several variants on handling the extra hour for the fall transition from EDT to EST. The MMS contains several macros to perform the conversion from the source system to the MMS standard which is Eastern prevailing time plus the appropriate time zone.

ERROR CHECKING

Error checking occurs after almost every SAS program statement. When the MMS detects an error condition, program execution stops, the job is marked as FAILED, and email notification is sent to the email distribution specified for the program.

- Database connections. The MMS checks that a successful connection to the database was completed by checking the SAS macro variable, SYSLIBRC.
- Database retrievals. The MMS checks that data is retrieved from the database successfully by checking the SAS macro variable, SQLRC. Optionally, the MMS can be directed to check for no rows returned from the query.
- Source metadata changes. Although there is a change management process, it is not uncommon for details of source table changes to occur without notification of downstream systems. The MMS maintains metadata about all of the production source tables from which it retrieves data. If there is a structural change in the source table, a warning email is sent and processing continues.
- After every data manipulation step. The MMS checks each resulting dataset to ensure that a dataset was successfully created, and optionally can check that the dataset contains rows.
- After every database update. When the MMS updates the database, the value of the SAS macro variable, SQLRC, is verified.

EMAIL NOTIFICATION OF JOB STATUS

The MMS frequently sends out email notification to inform the operation team of processing results. The email is always sent to a specified email group, but additional email addresses may be specified for each program. The MMU analysts may request that they be notified for certain programs. The MMS may send notification for the following:

- Successful program completion and a results summary may be included.
- Program failure, with a description of the cause.
- Changes in a table containing slow-changing dimensional data.
- Audit results. Some programs may expect a certain number of rows, or a certain number of input files. If the MMS does not find the expected information, it will process what it does find, but also emails a listing of missing information. This occurs most often for flat files.

JOB SCHEDULING AND WORKFLOW

The most significant enhancement of the MMS is its workflow architecture, which was added three years ago. Prior to the addition, job execution was managed only by setting the earliest start time for each job. The workflow architecture allows for definition of the relationships between jobs .The relationship between jobs is simple. Programs may have predecessors and/or successors. The information about the relationships is stored in the Oracle database, and the workflow processing is implemented in the kickoff program. The kickoff program will not select a program to run if all of its predecessors have not completed successfully.

Entries for all of the programs the MMS runs are listed in the job queue table. In advance, special MMS programs populate the job queue table with entries for each day. Each job entry contains the name of the program, the date range for which its SAS program is scheduled to run, the earliest time it can run and the current status. The status of each entry will change as the MMS runs each job: The possible statuses are:

- HOLD: The job's earliest run time has not passed, or the job's predecessors have not completed successfully.
- NOT RUN: The job is ready to be run and will be scheduled in an open program slot by the kickoff program.
- RUNNING: The kickoff program has spawned the MMS job.
- CANCELED: The MMS operator has specified that the job doesn't need to run.
- COMPLETE: The MMS job completed with no errors.

- FAILED: The MMS job has completed with errors.

When the job queue table is initially populated, each job entry's status is set to HOLD.

Some programs have no predecessors. When the current time is past the earliest run time, the kickoff program will switch the status of these from HOLD to NOT RUN.

If a program has predecessors, the kickoff program will use the status from the job queue table to determine if the predecessor programs for the corresponding date range has completed without errors. When this all predecessors complete without errors, the status of the corresponding job will change to NOT RUN.

The jobs will stay in the state, NOT RUN, until the kickoff program selects them to run. The kickoff program changes the state of the selected jobs to RUNNING, and then spawns each job. When the jobs complete, they update their completion status to COMPLETE or FAILED in the job queue table.

Occasionally, jobs require reruns. To rerun jobs, the status of a job and its corresponding successors is reset to HOLD and the kickoff program will execute the jobs as slots open.

CONCLUSION

By maintaining a simple framework and leveraging Base SAS® and SAS/ACCESS® products, the Market Monitoring System was built to populate the MMU data repository. Using SAS macros simplifies consistent program structure and design through code reuse. With the addition of workflow functionality, the MMS has proven to be scalable, now running more than 600 programs daily.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ellen Krawiec
Monitoring Analytics, LLC
2621 Van Buren Avenue, Suite 160
Eagleville, PA 19403
Work Phone: 610-271-8056
Fax: 610-271-8057
E-mail: Ellen.Krawiec@MonitoringAnalytics.com
Web: www.MonitoringAnalytics.com