Paper AD04

Trials and Tribulations of Automated SDTM Delivery

Edward Jones, PRA Health Sciences, Swansea, UK

ABSTRACT

Regular SDTM delivery is an ever more common request as sponsors and CROs move toward utilising SDTM or SDTM-like data structures to support operational activities, which in turn drives monthly and sometimes weekly or even daily SDTM generation and/or data transfers. Providing this service at an acceptable cost and to the required quality can be challenging. This paper plots the journey at PRA from a simple scripted approach following standard SDTM generation process (expensive and not scalable) through to the adoption of an enterprise scheduling tool driven methodology combined with streamlined SDTM generation process (lower cost, greater reliability). Specific common data issues that impact these deliveries will also be discussed along with their solutions/remediations for automated deliveries.

INTRODUCTION

In the beginning there was man….[1]. Then came CDISC standards [2], CDASH compliant eCRF designs and SDTM in particular [3]. We all know that CDASH compliant eCRF designs make producing SDTM datasets easy (easier). The idea that metadata can be leveraged to build SDTM compliant datasets is also largely accepted [4]. As an extension of this metadata-driven philosophy, the quest for a metadata driven eCRF build process is well underway.

Unfortunately, in terms of producing regular SDTM deliveries, vendor data transfers have proven tricky and continue to be a proverbial thorn in the side, particularly as trials more commonly gather biomarker and genetic data which can only be provided by small/niche vendors who have little or no knowledge of CDISC standards and frequently little or no desire to learn them! Many readers will have experienced the pain of trying to coach and/or cajole vendors into adopting CDISC standards, but after significant expenditure of time and effort the result has invariably been a disappointing lack of change in the vendor's delivery. Currently our approach at PRA is to take whatever the vendor is willing to send to us and just deal with it.

Nonetheless, various organisations have found the SDTM structure appealing (see What Good Are Regular SDTM Deliveries, below) and so demand has slowly grown over the years for ever more regular SDTM deliveries, as Steve Kirby *et al* put so nicely [4]:

What do we want? SDTM!
When do we want it? Now!
(Repeat as needed)
Ready? OK!

In response to the growing call for regular SDTM deliveries there has been a gradual increase in the automation of this process.

A HISTORY OF REGULAR SDTM DELIVERIES

Initially scheduled deliveries were still relatively infrequent and were provided via an entirely manual process. A Programmer would know that their client wanted an SDTM delivery on date XYZ, they'd log into to their SAS® environment, execute the necessary programs and probably manually upload the resulting SAS dataset after having eyeballed the log files and datasets. But, as the frequency of these deliveries reached monthly there was recognition

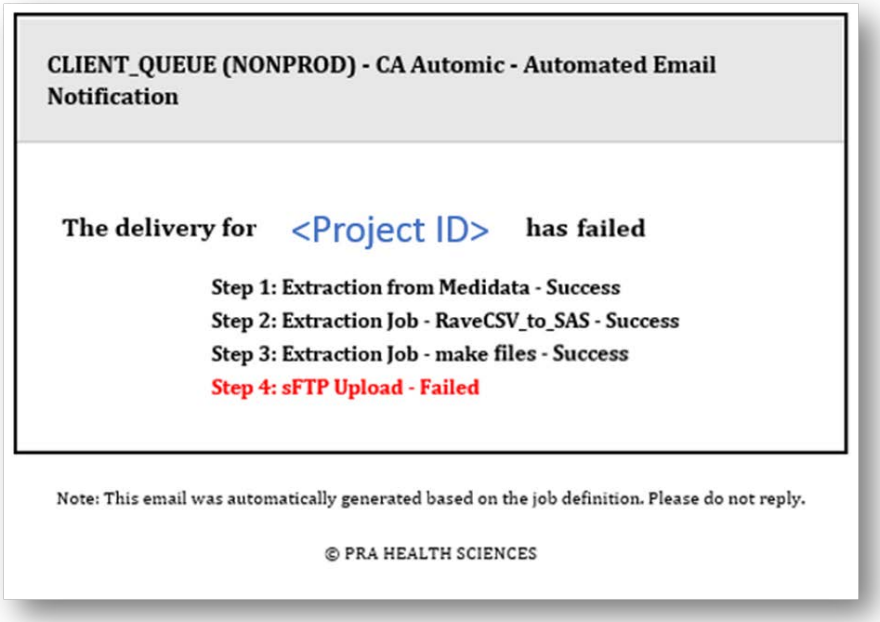
amongst Programmers (who have a predilection to automate things wherever possible) that some automation was called for, and a scripted delivery process was born. At PRA that meant DOS prompt scripts that could be scheduled via Windows Task Scheduler. Because these scripts were often first authored by Statistical Programmers who didn't necessarily have much experience in data integrations, these scripts started very simply, picking up source files from sFTP servers, executing SAS programs to transform the data according to the client's requirements and then posting the resulting datasets to a delivery location that was often another sFTP server. However, problems soon arose due to the simplicity of the procedural logic within these early scripts. In some cases it was possible for the SAS programs to begin execution before the source data had been successfully and fully transferred to the SAS working directory – not ideal. So, wait times were introduced into the scripts. But how long to set them for? Too long and the delivery process runs for ages. Too short and partial datasets might be delivered. Also, what if there's an error or warning in the log files of one of the scripted processes? Well, those log files can be attached to an e-mail and sent to the Programmer for review, but this isn't particularly practical once these deliveries start to scale upwards and a given Programmer is receiving tens of automated e-mails a month. As is possibly being demonstrated played out currently in the world of autonomous vehicles, human nature can mean that when we come to trust a system after seeing it perform successfully a number of times, the amount of attention we pay to monitoring that system may reduce, even when (in the case of autonomous vehicles) that lack of attention could result in imminent death [5].

Simple scripts are also not ideal for managing dependencies between steps. For example, if a failure occurs in the source data extraction step, that failure may not be detected by the script and the remaining scripted processes may well continue to execute. As a result, it becomes possible for the scripted process to make a delivery without having extracted any new source data and unless the log files were carefully read it might take some time before the issue is spotted.

ENTERPRISE SCHEDULERS – THE FUTURE

Enterprise schedulers have a number of key advantages over the simpler scripted approach and what's even better is that, if your organisation is reasonably large, the likelihood is that it already has one! So, if you've not already, go and have a conversation with your IT department using the following key words or phrases: enterprise scheduler, batch manager or batch scheduling, workload automation, middleware. This should at least get a useful conversation started. Refer also to the list of job scheduler software on Wikipedia [6]. Why try to make the world's most comprehensive scripted process when you can use an enterprise scheduling tool to do the heavy lifting for you?
Enterprise schedulers provide a number of advantages [6, 7, 8] above and beyond the simple scripted approach described earlier:

- Real-time integration of business activities across different operating system platforms and business application environments.
- Enterprise-wide view of all scheduled jobs / processes across multiple platforms / systems.
- They sequence scheduled processes that may have dependencies across multiple systems.
- Allows the Programming department to avoid dealing with day-to-day automation / scheduling issues.
- Provides the ability to manage scheduled job load on databases & files servers from one central location.
- One central location for sending and managing alerts as and when they occur.
- Dashboard view of what happened overnight.
- Built in (configurable) re-try functionality
- Can be event driven, thereby ensuring that jobs run when they are supposed to whilst decreasing the chance of errors caused by manual or timed jobs. Enterprise schedulers make it easy to automate and maintain your schedule.
- Managed dependencies can reduce the time it takes to run a schedule by 40 to 60 percent over manual or time-based scheduling. An event-driven schedule knows when a specified job completes, a file arrives, or a process ends, and immediately runs the next one. And, event-driven scheduling works across partitions and operating systems.
- One Scheduler to Rule Them All: Stick with a single, centralized automation solution. Eliminate SAS scheduler, Windows Task Scheduler, Cron, Task Scheduler, and SQLServer schedules and build your control into one place.
- Use your enterprise scheduler to read the log files of the various processes it runs and produce a consolidated success/failure e-mail that make it easy for the recipient to quickly determine the status of the job and review the associated log files if any elements did fail:



CLIENT_QUEUE (NONPROD) - CA Automic - Automated Email Notification

The delivery for <Project ID> **has failed**

Step 1: Extraction from Medidata - Success
Step 2: Extraction Job - RaveCSV_to_SAS - Success
Step 3: Extraction Job - make files - Success
Step 4: sFTP Upload - Failed

Note: This email was automatically generated based on the job definition. Please do not reply.

© PRA HEALTH SCIENCES

In PRA's experience, switching to an enterprise scheduler has yielded a far more stable and scalable scheduled delivery process. Issues are now much easier to spot, they're spotted far earlier, and they're occurring less frequently. The majority of the issues we now see are connectivity issues affecting the sFTP transfer to the client's server that result in the upload failing. These can typically be addressed by implementing retry functionality through your chosen enterprise scheduler. We have also encountered one file path configuration issue:

```
%let what1=…\Primary\Study Data\SDTM Datasets\XPT\\*.xpt;
```

This issue was identified and fixed on the first occasion that the job was run.

OTHER ELEMENTS TO CONSIDER

Having decided to do down the route of delivering SDTM datasets via an automated process, you're going to want to also start delivering them early in the life of the trial thereby making the trial data available to the consuming systems as soon as possible. In order to achieve this its necessary to get the SDTM programs written and tested as early as possible. You will, of course, have followed a metadata driven approach to develop your programs and datasets, but in order to test them prior to the availability of production data, it's desirable to establish clinically relevant dummy data. This dummy data should be clean (unlike the data that we receive in the real-world) because its primary purpose is to ensure that your programs work correctly when expected/anticipated data is received. In most cases the Electronic Data Capture (EDC) system used on the trial will undergo a UAT phase and its typically possible to ask your Data Management team (nicely) to enter some clinically relevant test data on your behalf. This approach has the advantage that having established that the programs work with clean data, it's possible to conclude with reasonable certainly that any subsequent errors or warnings encountered on production data relate to data issues and not a fundamental issue with first time programming. This in turn allows any resulting issue investigation to be completed initially by the data management team and not programming resource.

Dirty data can be the bane of a Programmer's life, particularly when regular SDTM deliveries are being made since it is unlikely to be practical for the trial's data to have been thoroughly cleaned before it must be included in an automated weekly SDTM delivery. Nor will it be possible for the programmer to code around all data issues as they arise since the volume and frequency of these issues is likely too great for the necessary programming effort to be implemented in a way that would meet the downstream consumer's cost expectations. Instead a discussion must be had with the consumer of the data, be they internal or external, and an agreement reached on the degree of quality control to be performed on the SDTM datasets. For example, subjects will frequently appear in vendor data before

that subject's information has been captured in the EDC system by the site. As such, a USUBJID may appear in LB, but be absent from DM (dependent on how DM is built). Whilst this is clearly not CDISC compliant, it may still be an acceptable scenario of the consumer of the SDTM datasets. Similarly, its highly likely that values will be received from vendors (and even sometimes through the eCRF) for which controlled terminology applies, but where the value entered is not compliant. In this scenario the consumer of the SDTM datasets is likely still willing to accept the non-compliant datasets and might well prefer to see the non-compliant value passed through into the SDTM rather than see it dropped due to its non-conformance with the defined controlled terminology for the field.

Having gotten your SDTM programs written and tested early, an agreed an acceptable approach regarding how dirty data will be handled, you probably also want to think about establishing a Service Level Agreement (SLA) with the consumer of your deliveries. An SLA is really just as relevant for a monthly transfer as it is for a daily transfer because expectations should be set on how often the deliveries might not run and how quickly any failures are likely to be dealt with. Particularly where frequent deliveries are concerned (weekly or daily) it's unlikely that your IT environment will be able to support uninterrupted deliveries 24 hours a day, 365 days a year. IT systems inevitably need maintenance. Software patches must be applied. Physical components replaced. These activities all need down time, which might well impact your scheduled SDTM deliveries. Additionally, there will be times when extractions from the source systems (e.g. the EDC system) may not be possible. For example, whilst an eCRF change is being applied. These sorts of activities are often targeted in periods when the trial users are less likely to be online, which is often a small window given the global nature of many trials. This is frequently the exact same window that you've scheduled your automated delivery jobs to run.

WHAT GOOD ARE REGULAR SDTM DELVERIES?

Having described how you might go about establishing a successful automated SDTM delivery process, why might such regular SDTM deliveries be useful? PPD have established a system – the Preclarus Patient Data Dashboard – which uses SDTM datasets in a TIBCO Spotfire® environment to provide various visualisations of the trial data that can be used to support areas such as safety review, identifying data issues, supporting dose escalation and identifying trends [9]. Similarly, Medimmune use SDTM in their data review processes, coupled with a visualisation tool [10]. In a slight twist upon this theme both Janssen [11] and Astra Zeneca [12] use a similar data review model, though this time based on a hybrid of CDASH and SDTM. The commonality across all of these, and other similar data review solutions, is the need for a consistent/stable data structure (SDTM or a similar variant) that enables the data from multiple trials to be consolidated into one place and where these data structures can be produced on a regular basis, ideally via an automated process.

CONCLUSION

Keeping track of regular SDTM deliveries via simple scripted processes is hard. Identifying and dealing with the issues that result from deliveries made via such simple processes is even harder. In all probability your organization already has software with enterprise scheduling functionality, so seek it out and start using it because doing so will make the setup and ongoing maintenance of regular SDTM deliveries so much easier, and everybody wants an easier life, right?

REFERENCES

1. The Wachowskis: The Animatrix, https://en.wikiquote.org/wiki/The_Animatrix
2. CDISC web site: Study Data Tabulation Model (SDTMIG). V3.2. http://www.cdisc.org/
3. Study Data Technical Conformance Guide: U.S. Department of Health and Human Services, FDA, CDER, CBER. https://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf
4. Steve Kirby, Mario Widel, Richard Addy: Building a Fast Track for CDISC: Practical Ways to Support Consistent, Fast and Efficient SDTM delivery PharmaSUG 2017 – Paper IB07
5. Thomas Claburn: Tesla driver killed after smashing intio truck had just enabled autopolit – US crash watchdog; https://www.theregister.co.uk/2019/05/17/tesla_autopilot_crash/

6. https://en.wikipedia.org/wiki/List_of_job_scheduler_software

7. Srinivas Manthena: Key Advantages of Enterprise Scheduling, https://smartbridge.com/key-advantages-enterprise-scheduling/

8. Pat Cameron: 12 Things You Should Know About Enterprise Job Scheduling, https://www.helpsystems.com/resources/guides/12-things-you-should-know-about-enterprise-job-scheduling

9. Philip C. M. Bartle: Visualising CDISC SDTM for Monitoring and Review; PhUSE 2017 DV05, https://www.phusewiki.org/docs/Conferene 2017 DV Papers/DV05.pdf

10. Deepika Sunkara, Jagadish Dhanra: Supporting A Data Review and Visualization Application With SDTM; PhUSE US Connect 2018 Paper DV09, https://www.lexjansen.com/phuse-us/2018/dv/DV09.pdf

11. Lieke Gijsbers, Tom Van der Spiegel: A proprietary, CDASH/SDTM-hybrid data model to expedite clinical data review; PhUSE 2018 Paper SI25, https://www.lexjansen.com/phuse/2018/si/SI15.pdf

12. Shyamprasad Perisetla, Suzanne Tautz: REACTing to data: The Use of Data Visualisation within Early Clinical Statistical Programming at AZ; PhUSE 2017 Paper DV03, https://www.phusewiki.org/docs/Conferene%202017%20DV%20Papers/DV03.pdf

CONTACT INFORMATION

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

Edward Jones

PRA Health Sciences
Llys Tawe, Kings Road, SA1 Swansea Waterfront,
Swansea, UK, SA1 8PG

Email: jonesedward@prahs.com
Web: prahs.com