

## SDTM to ADaM Programming: Take the Leap!

Jennifer McGrogan and Alyssa Wittle, Covance, Inc.

### ABSTRACT

Does ADaM programming ever feel intimidating or outside of your expertise? Then this presentation is tailored just for you. Why is it time to take the leap into learning more about ADaM or even to dive into ADaM programming? The more you understand about the path clinical data goes through to arrive at the analysis, the better a programmer you will be!

This presentation will highlight why it is important and exciting to learn about ADaM theory and ADaM programming. Where do you even start when learning about this topic? We will give some recommendations for effective ways to learn about ADaM and ADaM programming without getting overwhelmed in all the very technical details up front. We will give examples of common pitfalls and solutions often encountered when just starting out in ADaM. Finally, to assist in your success in this learning experience, we will give some tips and tricks to help you along the way.

### INTRODUCTION

Frequently when beginning work in the clinical trials industry a new statistical programmer will likely get their first experience by starting on a project programming Study Data Tabulation Model datasets (SDTMs). Even once comfortable in that area, learning the Analysis Data Model (ADaM) theory and getting experience programming ADaM datasets can be intimidating or seem out of reach. Additionally, with demanding timelines and impending submission dates, it can feel overwhelming to begin an ADaM project with little to no previous experience. Perhaps it is simply more comfortable to remain working with SDTMs and develop an expertise in this standard. However, what's important to keep in mind is that the more you understand about the path clinical data goes through to arrive at the analysis, the better programmer you will be.

It might seem monumental to decide to take on a new CDISC standard. So often there can be resourcing complications that lead to issues with getting training for ADaM datasets and receiving assignments that are ADaM-related at the same time. Even after finding solutions to those challenges, there is still the task of beginning to understand the Implementation Guide(s) and how to apply the concepts. Certainly, there is nothing wrong with perfecting SDTM knowledge and working to become an expert on the subject. However, it would be beneficial to the entire project team for the SDTM programmers to have an understanding of the needs associated with analysis and what might be expected from an ADaM perspective.

### ADVANTAGES

A simple advantage to team members having an understanding of both standards is that it can be helpful in avoiding rework at the SDTM-level. As the old adage goes, if you plan to succeed, you must succeed to plan. A thorough review of the tables, listings, and figures, and any other associated study documents is highly recommended at the onset of a study so the datasets do not need to be updated for later considerations such as custom naming of tests, the names of supplemental variables, or even organizing your data in a way that makes your analysis goals easier. It will be easier to foresee these hurdles and plan for them if you understand what is needed at an analysis level and why. Having this understanding will also assist in project team conversations with coworkers—it will be easier to communicate about the needs of the study and find the best way to meet end goals. If you better understand the needs of a project from an analysis level, this can facilitate brainstorming solutions and proposing problem-solving tactics.

Developing ADaM programming skills does not just expand one skillset, but it will also increase your value as an employee because you will have the versatility to work in either space. By practicing programming with both SDTM and ADaM, you are not restricted to specific portions of a project – you can work in all places. This allows you to be resourced to the same projects for longer; to spend more time with the same protocols, datasets, team members, indications and compounds; and to acquire a deeper understanding of the path of the data from collection through final analysis. Implementing this structure in study planning for team members who already have experience in both SDTM and ADaM can also provide an excellent learning environment for a new-to-ADaM programmer to get experience.

For instance, if a programmer is highly confident in their SDTM programming for Findings domains, an ideal resourcing plan for them could include programming a given domain at the SDTM level, and following that through to the associated ADaM in the same study, and potentially even further to programming the outputs related to these datasets. Allowing programmers to actually see values transformed through this pathway that the data takes will

enhance their understanding of the standards, the end goals, and ultimately improve their skills. It is possible for eager programmers to develop this experience across separate projects, but the ideal scenario is to give those programmers the end-to-end experience within a single project.

## TRAINING EFFECTIVELY

### MENTORING

As programmers develop their skills in new standards, it is critical that the learning be shared and appropriately disseminated to less experienced team members, because learning stops when teaching stops. An excellent solution to this challenge is to incorporate a mentoring/shadowing program. This is advantageous for two reasons. First, by providing a resource that the new-to-ADaM employee can ask questions and go to for help, they feel more confident tackling study assignments and gaining some experience. Second, the mentor will be challenged in exciting ways to find explanations and answers that effectively guide and instruct the new programmer. They will be forced to think about certain topics in new ways to provide possible solutions or suggestions for the pitfalls the new programmer encounters. This could help with team comradery as well, by providing priority resources for questions and comments. A team works more efficiently when they are communicating well and sharing learning experiences, whereas a new programmer who is not provided with a mentor might not be sure what questions are appropriate for whom. Ideally, once the new ADaM programmer gains some experience, and some confidence, they too can try their hand at mentoring another new programmer. By utilizing this mentor/mentee relationship successfully, knowledge can be more quickly and effectively shared. Sometimes the best learning comes from a painful programming mistake. It is, of course, advantageous to share these experiences with younger programmers so they can implement approaches that might help them avoid the issues altogether.

Furthermore, having a mentoring program in place can ease some of the angst that might accompany a brand-new programmer starting on a project for the first time. By providing a mentor as a resource who would be able to perform code review and assist on assignments, it will be easier to catch issues with enough time to fix them. It provides more time to update the code, but also to explain to the programmer what might be wrong with their understanding of a topic, and let them revisit trainings for weaker areas.

### “BITE SIZE” TRAININGS

There are different approaches to tackling ADaM for the first time. One popular option to administer ADaM training is for a Subject Matter Expert (SME) to provide a multi-day, basic overview training that covers the broader aspects of applying ADaM standards. While full of useful information, an 8-hour session (or even various 2-to-4-hour long sessions) can be too long for anyone to remain alert and engaged. Additionally, familiarizing yourself with the Implementation Guide (IG) will undoubtedly provide insight to the rules and standards, but as a stand-alone document it does not contain everything you need to know to succeed. As an alternative, ADaM concepts can be easier to digest when broken down into mini-trainings on some of the key aspects, each session lasting only about 15-20 minutes. Leaving time for the information to sink in and providing small exercises that go over the material can help solidify these mini-lessons and enhance the learning of any trainees. By tailoring these smaller trainings to what you are working on (for example, variables specific to the ADaM structure or dataset you are assigned to), they can be designed to explain particular aspects of the project, and when applied immediately, can reinforce the topics in a practical setting.

There are times when ADaM projects are not readily available in your company's current workload to provide real-world experience to new programmers. Although it can take time to prepare, a test area could be created with a “dummy” study that removes identifying information, and contains both SDTMs and ADaMs, to provide a similar experience. In this situation, the trainer would have the freedom to train on any portion of ADaM that they saw fit, and the options are endless! Mini-trainings could include an easily-digestible section on PARAM and PARAMCD, and be accompanied with training exercises that ask the programmers to create values for these variables based on study data. A training could be given on windowing, since this concept is exclusive to ADaM programming and an SDTM-specific programmer may not be familiar with it. These smaller trainings can be used to hone in on concepts that might require more explanation, and allow a space for programmers to ask questions or raise concerns. The flexibility allowed by these trainings being more “bite-size” can allow for meeting multiple times a week without taking up too much time or affecting billable project-work. By simulating a class-like environment with these mini-trainings and feedback exercises, you can ensure constant exposure to ADaM topics multiple times a week, even if project work may not be reinforcing these same concepts. Other mini-training topics could focus on any portion of the standard, for example date imputations, occurrence flags, and other timing variables. It is highly recommended to tailor these trainings to any application of the standard that programmers are struggling with.

## UNEXPECTED BENEFITS

Sometimes there could be added benefits that you did not even anticipate. For example, in my own training experience, there was a mini-training held that focused on the concept of windowing in ADaM datasets. While the word is thrown around a lot, without an in-depth training it can be hard to understand exactly what it is or how to apply it. The training consisted of about 45 minutes of information, using slides that had examples from statistical analysis plans (SAPs) and actual datasets. Once through the information, there was a time for questions for our small group to ask any clarifications we might have regarding the topic. Finally, it was accompanied by a small exercise in which we were provided analysis ranges from a SAP and then we populated the expected values in a dataset. Since I was not actively assigned to program any ADaM involving windows, I did not expect to be able to use the information very quickly. However, I was pleasantly surprised that while populating reviewer's guides for an upcoming ADaM submission, I was able to review the SAP and dataset specifications to populate the windowing information, rather than needing to reach out to the project's lead programmer for it.

## A NEW LEVEL OF PROGRAMMING

In addition to improving efficiency among the team, learning ADaM standards and implementing that knowledge in ADaM programming is an excellent way to personally improve SAS® programming skills. Due to the standardization of SDTM data, while some domains may contain complex mapping, it does not tend to require as much creative problem solving as programming ADaM does. Standard SDTM programming templates are likely much more detailed than the ADaM templates. This is not for lack of planning, there is simply much less that can be predicted in this more complex standard. Due to this, SDTM can provide an excellent starting point, but ADaM programming allows more room for development of programming skills and can help foster a deeper understanding of programming with SAS.

Considering the increased complexity of ADaM programming, personal development goals can be achieved in many different ways. You will use merging techniques, functions and procedures which you likely had never been exposed to in SDTM, such as complex transposes to determine sums and totals, deriving new records, and computing complex calculations and flags. Until you start programming in ADaM it is hard to anticipate just how many interesting programming strategies you will have to use! Additionally, you may be faced with situations similar to those seen in SDTM, but much more repetitive. Where simple conditional programming may have been appropriate at an SDTM level, it could be much more time-intensive to program it with the same strategy, so creating a macro could be a solution. By exposing yourself to the programming opportunities laid out in ADaM specifications, you are expanding your ability to implement these new-to-you SAS functions on actual trial data, and gain a deeper understanding of how and when to use them.

## WHERE TO BEGIN

Now that you are convinced that learning ADaM programming can enhance your career as a programmer, you are likely wondering where to begin. A basic explanation given to programmers getting started in the industry has remained with me as I continue to program and work with each standard. "SDTM is input-driven and ADaM is output-driven." Having an understanding of this is the first step to working with these standards. We can expand on the explanation further—SDTM is a standard structure that was created for tabulating data in a consistent way across submissions, while ADaM is a standard structure that was written for deriving and analyzing datasets to output and support specific analysis results. Understanding this core difference, and knowing the rules are centered around these goals will help make sense of the differences you will find when transitioning from SDTM to ADaM.

## ASSIGN DATASETS EFFECTIVELY

Something else to consider when setting yourself up for success in programming ADaMs is beginning in the right place. It can, understandably, be intimidating to start something new and different. However, there are some methods to ease some of those feelings, like beginning with the right dataset. With ADaM dataset possibilities being so broad, it is easier to recommend where not to start than where to begin. ADSL is one of the most complex ADaM datasets and needs to be programmed before other dataset programming can really begin. However, it incorporates several source SDTM datasets and can be one of the most difficult to understand and program, so it is not an ideal starting point for someone trying to get their first ADaM experience. Similar notes can be made about ADLB or the efficacy datasets. It would be best to choose a subject area where the programmer is already comfortable with the SDTM—be that adverse events data, exposure data, medical history, etc. For training purposes, the most effective experience would likely be programming the associated ADaMs for those areas. Ideally, the programmer would progress to the table/listing/figure programming and assist in creating those outputs as well, to gain a strong understanding of the path of the data in the project. In general, a basic dataset structure (BDS) ADaM would be a good place to start as projects tend to have datasets that support simpler BDS domains such as vital signs and electrocardiograms. Any ADaM dataset has the potential to be complex, but these BDS examples would likely be the most digestible first

experience to key variables and the dataset structure.

As a programmer gains more skills and confidence, they can progress to more complex occurrence datasets (often including imputations) or other more highly derived BDS datasets and eventually ADSL. Finally, a last piece of advice is to consider what experience you have and what might complement that well. I entered this industry as an intern, and have since grown my skills and abilities to become a competent programmer. When I first began, due to need, most of my assignments were related to creating SDTM-compliant define.xmls. It was a natural progression for me to start working on ADaM-compliant define.xmls as I gained more experience. By getting my feet wet in that manner, I was able to get comfortable with ADaM standards from a define.xml position, piquing my interest and making it easy for me to pivot into ADaM programming assignments.

## REVIEW STUDY DOCUMENTS AND UNDERSTAND THE GOAL

Given the suggestions and recommendations I have provided; I hope I have inspired some excitement for programming in the ADaM standard. However, when you receive your first ADaM specifications and are ready to begin your programming assignment, there are a few things to consider before diving in. Of course, you will want to keep in mind your company's standard processes and good programming practices. In addition to that, you will want to form a strategy for attacking the specifications. You will first want to have a clear understanding of the study goals and objectives. Read the SAP thoroughly, and review the associated shells and templates that may have been provided. Remember, ADaM is driven by the output, so you will want to be very familiar with the results you are trying to produce. If your company practices annotating the tables with variables, be sure to review those against the specifications you were provided. You will want to be able to identify the variables that will be used and understand the functions they will have to go through to get there. In my experience, some of our beneficial mini-training assignments included reviewing table outputs and providing what variables and procedures might be necessary for producing the output. Recall that there are many ways to reach the same goal – but you should pick one and have that plan in mind.

## CREATE A PLAN OF ATTACK

In general, a good strategy for beginning a program is as follows:

- Gather the following files:
  - o A copy of the starting template you will be using, if you have one
  - o A copy of any of the outputs related to the dataset you are programming for
  - o A copy of any relevant implementation guides
    - The ADaM IG only contains ADSL and BDS structures, while OCCDS structure is in a separate document, and Therapeutic Area User Guides (TAUGs) are each individual files as well, be sure to collect each file which is applicable to your assignment
- Create a list where you have categorized the variables according to dependency.
  - o Similar to SDTM programming, some variables need to be created in a certain order to derive other variables. Have an understanding of which variables require others to already be created, and consider this in your programming.
- Retain any SDTM predecessor values.
- Program basic assignments such as creating analysis dates or formatting PARAM/PARAMCD from xxTEST/xxTESTCD.
- Program in order of dependencies - If you are programming a variable to represent the change from baseline (CHG), you will first need to populate the result values (AVAL), the baseline flag (ABLFL), and the baseline value (BASE). You will then compare the baseline value to the specified result to determine the change. By keeping variable dependencies in mind, and visualizing what the outcome of the variables should actually look like, you will have a more concrete programming goal which is easier to achieve.

Consider other programming advice as you work as well. It can be helpful to speak your strategy out loud – whether bouncing ideas off of a coworker or just verbalizing your thoughts, stating your plan out loud can help you fine-tune your plan or determine flaws in logic. Collaborating and sharing programming ideas can give you some of the most useful feedback you can find. Do not back down from programming challenges with new functions or procedures either. As stated above, ADaM programming can allow you to expand your programming abilities based on the needs of the data. This is where that magic happens—try merging data in new ways, using proc transposes, or applying arrays for the first time. Trial and error is how you will learn, so do not be afraid to try new things.

## CONCLUSION

Hopefully you are eager to take the leap and finally get that ADaM programming experience you may have been apprehensive about! We have provided strategies for how to implement trainings to make programming ADaM seem more achievable. Excellent resources exist on the CDISC website as well, where there are downloads of all current and draft versions of the different IGs and other helpful documentation. Parts of the guide can be difficult for a novice to interpret but as more time is spent with it, and as more programming experience is gained, you will develop an understanding of how the guide is organized and be able to apply it to topics with ease. To paraphrase Theodore Roosevelt, “Nothing worth having comes easy” and this certainly is true of becoming a great ADaM programmer. However, with a little patience, preparation, and practice you may surprise yourself with how advanced you become!

## CONTACT INFORMATION

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

Jennifer McGrogan  
[Jennifer.McGrogan@Covance.com](mailto:Jennifer.McGrogan@Covance.com)

Alyssa Wittle  
[Alyssa.Wittle@Covance.com](mailto:Alyssa.Wittle@Covance.com)