PharmaSUG 2019 - Paper AD-299

Best Practices for ISS/ISE Dataset Development

Bharath Donthi, Lingjiao Qi, Statistics & Data Corporation

ABSTRACT

The integrated summary of safety (ISS) and integrated summary of efficacy (ISE) are vital components of a successful submission for regulatory approval in the pharmaceutical industry. ISS and ISE allow reviewers to easily compare individual outcomes, tracking subjects' results across the entire clinical development lifespan of the investigational product. Furthermore, ISS/ISE facilitate broad views of the investigational product's overall efficacy and safety profiles. However, building integrated datasets is a challenging task as it requires the programmer to achieve consistent structures and formats while also ensuring that each dataset is CDISC-compliant.

This paper provides best practices for ISS and ISE dataset development to guide integrated analysis dataset design and production in an efficient manner. First, we discuss best practices to ensure the consistency of integrated datasets by up-versioning all data with the same coding dictionaries (MedDRA, CTCAE, WHO, etc.) and by harmonizing all variable attributes (variable names, types, formats, labels, CODE and DECODES for categorical and ordinal variables, ranges for continuous variables, etc.). Next, we discuss CDISC requirements regarding the mapping of SDTM and ADaM. Then, we will talk about how to handle some complex cases in developing integrated datasets, such as when one subject participates in multiple clinical studies included the ISS/ISE. Finally, we will touch on key points of analysis involving consistent flag assignment across studies and proper application of integration methods for safety and efficacy analysis. This step-by-step guide enables the efficient and accurate creation of ISS and ISE datasets.

INTRODUCTION

The ISS and ISE are required by the U.S. Food and Drug Administration (FDA) as a critical component in any New Drug Application (NDA) submission. Specifically, the FDA's Guidance for Industry Integrated Summaries of Effectiveness and Safety states,

> "The ISE and ISS are not summaries but rather detailed integrated analyses of all relevant data from the clinical study reports that belong in Module 5. We consider the ISE and ISS critical components of clinical efficacy and safety portions of a marketing or licensing application. Therefore, the ISE and ISS are required in applications submitted to the FDA in accordance with the regulations for NDA submissions."

Figure 1 illustrates the submission structure based on Module 5 in the Electronic Common Technical Document Structure Defined by FDA. Both ISS and ISE are integrated documents describing the results of multiple clinical trials where the results of individual clinical studies are combined into one integrated database and are summarized together. Thus, ISS and ISE offer insight beyond what is observable in individual clinical trials.

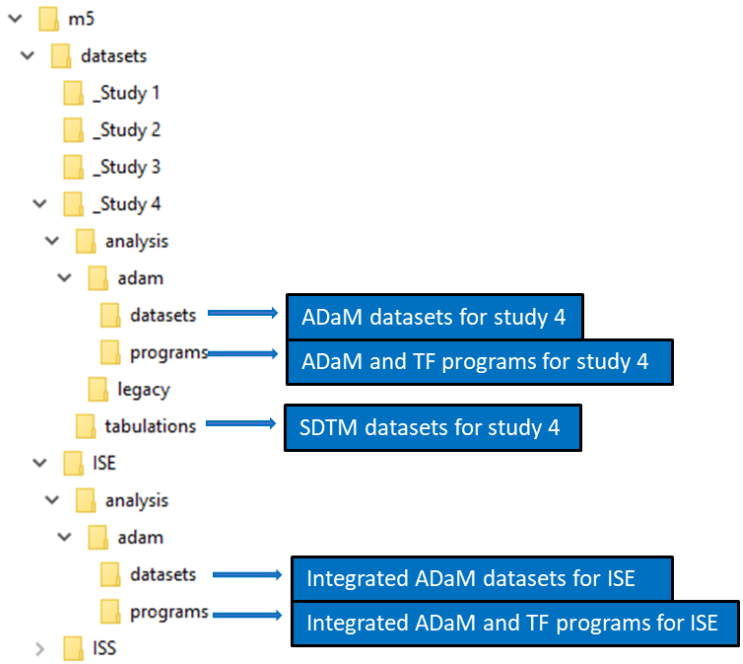

Figure 1: Module 5 Electronic Common Technical Document Structure Defined by FDA

Though ISS and ISE are both important for e-submission to FDA for drug approval, the production of a database containing the combined study data is lengthy and challenging for the programmers working on the integrated summaries. The purpose of this paper is to provide best practices for ISS and ISE dataset development to guide integrated analysis dataset design and production in an efficient manner. This paper assumes the audience has basic knowledge of data flow in the clinical industry and some experience with CDISC standards.

INTEGRATION STRATEGY

Before any programming activities, the sponsor should evaluate and determine which studies will be part of the submission. The study statistician should then detail each study to be pooled for the ISS/ISE in an integrated statistical analysis plan (SAP) for Safety or Efficacy. The SAP should also include a list of integrated analysis tables, listings, and figure (TLF) outputs, whose mock-ups should be provided as well. Once the scope of the integrated analysis is clear and supporting documents (electronic case report forms [CRFs], datasets specifications, etc.) are available for each individual study, programmers can start to plan and design integration datasets. Since the FDA mandate for CDISC submission started in 2017, this paper assumes all individual studies have already been converted to CDISC format.

Integrated datasets for ISS/ISE can be built in several different ways. One way is to build integrated analysis datasets through integrated SDTM datasets (Figure 2). The first step is to merge the SDTM parent domain with the corresponding supplemental domain (if any) by supplemental qualifiers for each individual study. The second step is to stack these processed SDTM domains together to create integrated SDTM datasets. Integrated ADaM datasets can then be created based on the integrated SDTM domains. This approach may make it easier to spot inconsistencies between studies, but it can be very time-consuming.

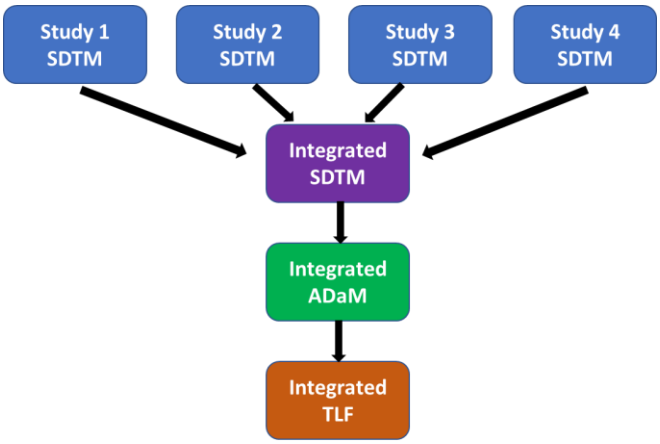

Figure 2: Build Integration Datasets via Integrated SDTM Datasets

A more popular and simple option is to directly build integrated analysis datasets using individual ADaM datasets from individual studies (Figure 3). After the programming work is completed for each individual study, integrated analysis datasets can be created by setting individual ADaM datasets together via a simple stacking program.

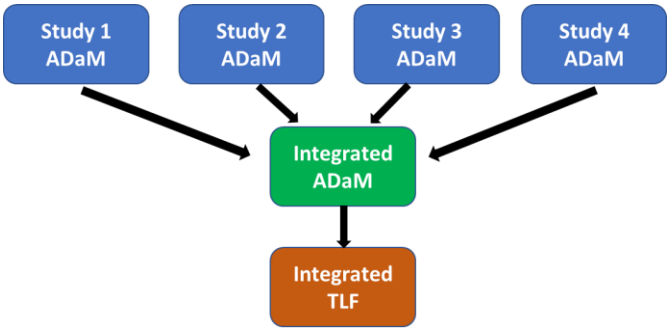

Figure 3: Build Integration Datasets via Individual ADaM Datasets

A possible drawback for this approach is that programmers need to wait until the individual study's ADaM datasets are programmed and validated. However, this approach offers flexibility to handle inconsistencies between studies in the stacked analysis datasets. Because ADaM standards are more flexible than SDTM standards, it is recommended that data inconsistencies between studies be handled when integrated analysis datasets are being built. For example, ADaM datasets may have different structures or content due to differences in the originating contract research organization (CRO), inconsistent study designs, customized databases, or differing analysis needs. Each inconsistency may be addressed during the ADaM stacking stage by updating or adjusting the individual datasets. Figure 4 illustrates an example where some SDTM information is added into an individual ADaM dataset before stacking with other ADaM datasets.

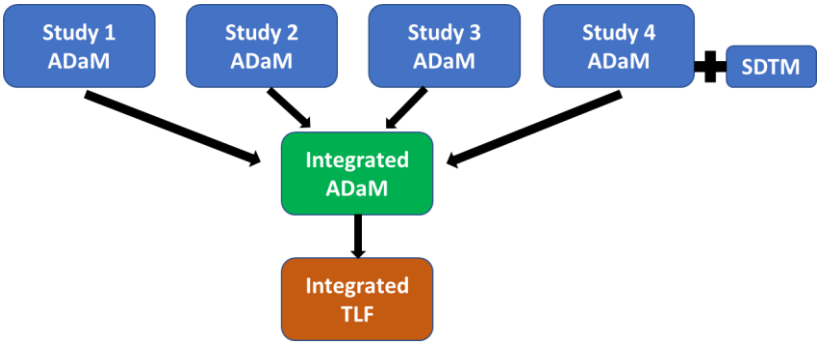

Figure 4: Build Integration Datasets via Individual ADaM Datasets with Flexibilities

Integration efforts are greatly reduced when all individual studies have a similar design and the datasets have consistent CDISC structure, though this is very rare in reality. Often, individual studies have different study designs and their databases are structured differently (e.g., dose escalation study vs. double-blind study). Additionally, the same variables in different studies might have different attributes, which makes integration programming activities challenging and time consuming. In the following sections, we will cover a broad list of best practices when developing integration datasets.

UPVERSION CODING DICTIONARIES

The first thing to do before starting any programming is to up-version all dictionaries to ensure a single version is used and to avoid data differences arising from version mismatches. Since new versions of coding dictionaries are released every few years, it is very likely that not all individual studies used in the integration are coded with the same version of dictionaries. For submission, it is required that all studies use the same version of the Medical Dictionary for Regulatory Activities (MedDRA) to code the adverse events pooled in the integrated analysis adverse events datasets. Similarly, the same version of WHODrug should be used to code concomitant medications pooled in the integrated concomitant medication analysis dataset and the same version of Common Terminology Criteria for Adverse Events (CTCAE) should be used for lab toxicity grading. Most of the time, reconciling coding dictionary versions is completed by coding specialists in the data management department before data is pooled to create integrated ISS/ISE datasets. If the version of the coding dictionary used for an individual study is different from the version of the coding dictionary used for the ISS/ISE integrated datasets, a document stating any changes to preferred terms or the hierarchy mapping should be provided to the FDA to help the reviewers understand any differences when reviewing pooled data against individual study data.

HARMONIZE VARIABLE ATTRIBUTES

Assuming individual study datasets all have the same dataset structure and all coding dictionaries across all studies are up-versioned with a single version, the next step is to make sure variables with the same name across individual studies have the same attributes before stacking them together. Checking and reconciling attributes is important for preventing future programming errors and warnings. Checks on variable types, formats, labels, length, etc. should be performed. Below is an example showing differences in the length of DSTERM in ADSL across four individual studies (Figure 5). Attribute checking can be done easily with a PROC CONTENTS and PROC COMPARE procedure in SAS.

Study	Variable Name	Variable Length
Study 1	DSTERM	$160
Study 2	DSTERM	$200
Study 3	DSTERM	$180
Study 4	DSTERM	$200

Figure 5: Variable with Same-Name but Different Length

ENSURE CONSISTENT PROGRAMMING LOGIC IS USED ACROSS STUDIES

Once inconsistencies in variable attributes are identified and harmonized, programmers need to make sure variables with the same names contain the same content before stacking individual datasets together. Different variable content could result in severe programming errors as they present different meanings. Figure 6 provides an example of inconsistent programming logic used when deriving the same variable, ANL01FL, in the efficacy dataset, ADEFF1, across all individual studies. Though all four studies have the same variable named ANL01FL in dataset named ADEFF1, the algorithm used is different. Compared to Study 1, Study 2 ANL01FL has one more filter when DTYPE = "" while Study 3 has one more filter when CHG is not equal to the missing value (CHG^=.), and Study 4 has yet another derivation where XOLOC is sorted differently from Study 1. Before stacking ADEFF1 from these four individual studies to create an integrated ADEFF1, the programmer and study statistician should agree to apply one consistent programming logic to derive ANL01FL and discuss how to update any other needed derivations.

In the example shown in Figure 6, there is no specific description assigned to the variable label of ANL01FL, which is very common in ADaM datasets. Without a specific description in the label, it is impossible to understand what ANL01FL signifies by looking at the labels in SAS. Therefore, we always recommend using descriptive variable labels for analysis flags or critical variable flags for all clinical studies. CDISC ADaM Implementation Guide 1.1 allows descriptive texts to be added at the end of the labels of variables whose names contain indexes "y" or "zz" (Figure 7 and Figure 8). Adding descriptive texts into these labels in individual study datasets will make integration tasks much easier for future programmers developing integration datasets. The programmer working on integration datasets should verify with the project statistician whether each sequential iteration of ANL01FL is intended to be applied to the same type of analysis.

Study	Variable Name	Variable Label	Programming Notes
Study 1	ANL01FL	Analysis Flag 01	Consider records where ATPTN>0 . And sort by USUBJID AVISITN CHG ATPTN descending XOLOC and assign Y for the last record within each visit
Study 2	ANL01FL	Analysis Flag 01	Consider records where ***DTYPE***=' ' and ATPTN>0 . And sort by USUBJID AVISITN CHG ATPTN descending XOLOC and assign Y for the last record within each visit
Study 3	ANL01FL	Analysis Flag 01	Consider records where ATPTN>0 and ***CHG***^=. And sort by USUBJID AVISITN CHG ATPTN descending XOLOC and assign Y for the last record within each visit
Study 4	ANL01FL	Analysis Flag 01	Consider records where ATPTN>0 . And sort by USUBJID AVISITN CHG ATPTN ***ascending*** XOLOC and assign Y for the last record within each visit

Figure 6: Example of Same-Name Variables with Differing Derivation Logic

CDISC ADaM Implementation Guide Version 1.1

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ANLzzFL	Analysis Flag zz	Char	Y	Cond	ANLzzFL is a conditionally required flag to be used in addition to other selection variables when the other selection variables in combination are insufficient to identify the exact set of records used for one or more analyses. Often one ANLzzFL will serve to support the accurate selection of records for more than one analysis. Note that it is allowable to add additional descriptive text to the label (see Section 3.1.6, Item 1).

Figure 7: ADaM IG Indicates Descriptive Text is Allowed to be Added to the Label of ANLzzFL

CDISC ADaM Implementation Guide Version 1.1

3.1.6 Additional Information about Section 3

In general, the variable labels specified in the tables in Section 3 are required. There are only two exceptions to this rule:

1. Descriptive text is allowed at the end of the labels of variables whose names contain indexes "y" or "zz"; and
2. Variable labels containing a word or phrase in brackets, e.g. {Time}, should be replaced by the producer with appropriate text that contains the bracketed word or phrase somewhere in the text (e.g., the label for a *TM variable is indicated as {Time} in this document) indicating any producer-defined label is permitted as long as the word Time is incorporated in it.

Figure 8: ADaM IG Indicates Descriptive Text is Allowed to be Added to the Label of ANLzzFL

When checking consistency on categorical variables, a simple PROC FREQ procedure will help to determine if special attention is required when combining data in integrating all datasets. For example, is ADAE.AECAT capitalized in all individual studies? Do the classified categories in LBSCAT have the same wording, spacing, abbreviations, capitalization, etc.? More importantly, does the same variable with the same value have the same meaning? Figure 9 presents an example showing inconsistency of the relationship between AVISIT and AVISITN. Before stacking the individual datasets together to generate integrated analysis datasets, the value of AVISIT when AVISITN is equal to 4 in Study 2 needs to be adjusted to be consistent with Study 1, and AVISITN in Study 3 and Study 4 needs to be reassigned to ensure data consistency across all studies.

AVISITN	Study 1 - AVISIT	Study 2- AVISIT	Study 3- AVISIT	Study 4- AVISIT
0	Screening	Screening	Screening	
1	Day 1	Day 1	Day 1	Screening
2	Day 2	Day 2	Day 3	Day 1
3	Day 3	Day 3	Day 4	Day 2
4	Day 4	Day 4 – Follow up	Follow – up	Day 3
5				Day 4

Figure 9: Different Meanings for Same Variable and Value across Studies

CONFIRM CDISC COMPLIANCE

Before being pooled to create integrated analysis datasets, the SDTM and ADaM datasets in each individual study should have undergone conformance checks (i.e., Pinnacle 21) with CDISC standards. Any warnings and errors found in the conformance check report should have been properly addressed. Unaddressed warnings and errors should be documented in the study data reviewer's guide or analysis data reviewer's guide. When SDTM data from individual studies are directly used to generate integrated analysis datasets, SDTM mapping checks across all studies are needed to identify any potential mapping differences which might be the result of multiple vendors being used, different database structures, different versions of the SDTM IG used, etc. Programmers can first check if the domains needed for

integration are presented in all studies and if their mapping contents are consistent across all studies. If supplemental domains are used for integrated datasets, programmers should also check if the supplemental qualifiers have the same QNAM and QLABEL. Domain checks and variable mapping checks should also be applied to individual ADaM datasets.

One of the CDISC rules for ADaM is the one-to-one mapping requirement for designated variable pairs. After stacking all individual ADaM datasets together to create the integrated dataset, it is worthwhile to check all CODE and DECODE variables to ensure they maintain a one-to-one mapping. Reusing Figure 9 as an example, AVISITN and AVISIT are paired variables which have a one-to-one relationship in individual studies, but the one-to-one mapping relationship fails in the integrated analysis dataset after stacking individual dataset together. Thus, further adjustment needs to be applied when programming integrated datasets.

Programmers should pay special attention to SITEGRy/SITEGRyN, AGEGRy/AGEGRyN, TRTxxP/TRTxxPN, etc. to ensure compliance to the one-to-one mapping rule in integrated datasets. In addition to this one-to-one rule, the integrated analysis datasets should also be compliant with other CDISC standards. It is recommended to run CDISC conformance checks on the integrated datasets during the development stage instead of waiting for all datasets to be produced. Running conformance checks in advance allows enough time to address errors or warnings.

COMPLEX CASES IN INTEGRATED DATASET STRUCTURE

Assuming individual study datasets all have the same dataset structure and each subject has only been enrolled in one study, individual ADaM datasets can be stacked together to produce the integrated datasets. In this case, the integrated ADSL has a structure of one record per unique subject, in compliance with CDISC ADSL rules. Other ADaM datasets (ADAE, ADCM, ADVS, ADLB, ADEF, etc.) could be stacked in similar fashion as well. However, in complex cases where subjects are enrolled in multiple trials, a simple stacking might not work. For example, when one subject participates in multiple studies, the first dose date might not be the first dose date in any one individual study. Since one subject is in multiple phases after stacking all data from individual trials together, it raises many difficult issues. How does one handle study periods? How does one define the first exposure date and treatment-emergent adverse events? To address these issues, there is a need to create new variables in the integrated analysis datasets.

The CDISC ADaM integration sub-team recently developed and released Version 1.0 of the standards document entitled “ADaM Data Structures for Integration” to provide guidance on these types of challenges in integration analysis. At the time of this paper, the draft document is available for public review and comment. The document allows for multiple records per subject in IADSL with the most basic structure being a one-record-per-subject-per-pool structure. With new variables POOL and POOLC, first exposure date and other baseline flags can be easily derived for patients participating in multiple trials. It also presents how integrated basic data structure (IBDS) and integrated structure for occurrence data (IOCCDS) will work effectively with the new IADSL class. Though only the draft version of this ADaM integration standards document has been released at this moment, we recommend readers review it as it sheds light on handling complex cases in developing ISS/ISE datasets.

OTHER POTENTIAL ANALYSIS ISSUES

The SAP for ISS/ISE should list details on statistical methods and statistical analysis rules for developing integrated analysis datasets and programming integrated TLFs. For example, the SAP should clearly define integrated analysis treatment groups and analysis populations. Programmers should find answers to questions such as: should patients dosed once a day be combined with patients dosed twice a day if the total amount of treatment is the same? Should all placebo groups be pooled together, even if the dose amount is different? The SAP should also provide clear direction on baseline flag definitions, as it is possible that baseline flags defined in individual studies are not derived consistently or the definition used in individual studies is not needed for integrated analysis, especially when complicated study designs are involved and one subject participates in multiple studies. The SAP should address whether unscheduled visits are included in integrated analysis.

The SAP should also detail how to handle missing data if imputation is needed. The most common imputation may be partial date or missing date imputation. Different logic might be used for partial/missing start date imputation when compared with partial/missing end date imputation. For example, a partial start date with a value of 2018JAN might be imputed as 2018JAN01, while a partial end date with a value of 2018JAN might be imputed as 2018JAN31. Missing values for analytical endpoints are often imputed as well. Common imputation methods are last observation carried forward and multiple imputation. When stacking individual ADaM datasets to generate integrated analysis datasets, there is no need to re-impute records with values imputed with last observation carried forward in individual studies, but values imputed with a multiple imputation method may need to be re-imputed as the missing patterns may have changed with multiple individual datasets stacked together. Programmers should refer to the integrated SAP and discuss with the project statistician to ensure proper imputation methods are used to derive missing values for integrated datasets.

CONCLUSION

ISS/ISE are vital components of a successful submission for drug approval in the US. In ISS/ISE preparation, the most challenging and time-consuming task is the production of the ISS/ISE analysis datasets containing the combined study data. This paper provides specific tips and techniques to efficiently develop integrated analysis datasets, including instructions for ensuring variable attribute consistency and advice for complex cases and potential analysis issues. These recommendations and best practices will help enable efficient and accurate creation of ISS and ISE datasets.

REFERENCES

Wayne Zhong, Kimberly Minkalis, Deborah Bauer, 2018 “ADaM Structures for Integration: A Preview” *PharmaSUG 2018.*

CDISC ADaM Data Structures for Integration Version 1.0 (Draft) “ADaM Data Structures for Integration: General Considerations and Model for Integrated ADSL, Integrated OCCDS, and Integrated BDS”

FDA. “Guidance for Industry Integrated Summaries of Effectiveness and Safety: Location Within the Common Technical Document”. *2009*

Tracy Sherman, Brian Fairfield-Carter, 2018 “ADaM Integration for Summary of Clinical Safety: The ‘Unique Patient’ Paradox” *PharmaSUG 2018*

Rajkumar Sharma 2015 “Tips on Creating a Strategy for a CDISC Submission” *PharmaSUG 2015*

Rajkumar Sharma 2012 “Creating an Integrated Summary of Safety Database using CDISC ADaM : Challenges, Tips and Things to Watch Out” *PharmaSUG 2012*

Balaji Ayyappan 2012 “CDISC SDTM CONVERSION IN ISS/ISE STUDIES: TOOLS” *PharmaSUG2012*

Changhong Shi, Qing Xue 2010 “Integrated Summary of Safety and Efficacy Programming for Studies Using Electronic Data Capture” *NESUG 2010*

ACKNOWLEDGMENTS

The authors thank Kevin Uchimura, Lot Slade, Kirk Bateman, Faith Kolb, and Melanie Ciotti at Statistics and Data Corporation for their thoughtful review of this manuscript.

CONTACT INFORMATION

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

Name: Bharath Donthi
Enterprise: Statistics & Data Corporation
Address: 63 South Rockford Drive, Suite 240, Tempe, AZ 85281
E-mail: bdonthi@sdcclinical.com
Web: https://www.sdcclinical.com/

Name: Lingjiao Qi
Enterprise: Statistics & Data Corporation
Address: 63 South Rockford Drive, Suite 240, Tempe, AZ 85281
E-mail: lqi@sdcclinical.com
Web: https://www.sdcclinical.com/