PharmaSUG 2014 - Paper DS07

Applying ADaM BDS Standards to Therapeutic Area Ophthalmology

Songhui ZHU, A2Z Scientific Inc., East Brunswick, NJ

ABSTRACT

Most of the ADaM datasets are in BDS data structure. So statistical programmers spend a lot of time implementing BDS datasets. In this paper, the author's thoughts and practices are illustrated using examples about how to implement ADaM-compliant BDS datasets when facing some complex situations in ophthalmology studies considering that each subjects can have 0 or 1 study eye and 1 or 2 fellow eyes, that the measurements can be performed on various segments of each eye, and that measurements can done at different time points.

1. INTRODUCTION

Most of the ADaM datasets are in BDS data structure. In general, almost all efficacy datasets and most safety datasets can be implemented in BDS data structure. Therefore, statistical programmers spend a lot of time implementing BDS datasets. However, implementing ADaM compliant BDS datasets are not straightforward. For ophthalmology studies, due to the facts that each subject has two eyes and that the measurements can be performed on both eyes, even on different segments in both eyes, the implementation of CDISC-compliant datasets is not straightforward. For SDTM, variables --SCAT and --LOC can be used to capture the information about eye and segment of the eye on which the measurements are performed. However, the corresponding ADaM implementation is not as simple as it appears. In this paper, the author will discuss some typical situations in Ophthalmology studies that need be carefully taken care of. Examples will be used to illustrate how to implement ADaM-compliant BDS datasets. The situations discussed are: 1) implementation of PARAM/PARAMCD for ophthalmology measurements; 2) implementation of PARCATy for ophthalmology measurements; 3) implementation of by-visit baseline; 4) implementation of by-time-point baseline; 5) implementation of multiple baselines; 6) implementation of lab data using two sets of units; 7) implementation of LOCF.

2. Creating ADaM-Compliant BDS Datasets

2.1 Implementation of PARAM/PARAMCD/PARAMN

For Ophthalmology studies, the measurements such as Schirmer's test and tear beark-up time (TBUT) are performed for both left eye (OS) and right eye (OD), respectively. Further, some measurements such as Fluorescein Comean Staining Score (FCS) and Lissamine Green Conjunctival Staining Score (LGCS) are performed for several locations such as nasal, temple, upper, lower, and central of the left eye and right eye, respectively. In these cases, implementing PARAM/PARAMCD/PARAMN is not as simple as copying SDTM variable --TEST/--TESTCD.

In SDTM datasets, the above information can be stored by the combinations of variables --SCAT, --TEST/--TESTCD, and --LOC. Therefore, the SDTM can be as follows:

Table 1. SDTM Domain XO

USUBJID	XOSCAT	XOTEST	XOLOC	XOORRES	VISIT
001-101-01	OS	Fluorescein Comean Staining Score	Temporal	5	Visit 2
001-101-01	OD	Fluorescein Comean Staining Score	Temporal	4	Visit 2
001-101-01	OS	Fluorescein Comean Staining Score	Nasal	3	Visit 2
001-101-01	OD	Fluorescein Comean Staining Score	Nasal	4	Visit 2
001-101-02	OS	Fluorescein Comean Staining Score	Temporal	3	Visit 2
001-101-02	OD	Fluorescein Comean Staining Score	Temporal	2	Visit 2
001-101-02	OS	Fluorescein Comean Staining Score	Nasal	4	Visit 2
001-101-02	OD	Fluorescein Comean Staining Score	Nasal	3	Visit 2

In ophthalmology studies, it is often that, for every subject, at most one eye is selected as the so-called study eye. The study eye is usually defined as an eye that meets some qualification conditions and must be treated with the study drug. The eye that is not study eye is called a fellow eye. Therefore, a subject may have 0 or 1 study eye and 1 or 2 fellow eyes. Most of the efficacy analyses are done for study eye and fellow eye, respectively. In order to implement the corresponding ADaM datasets, one need decide how to implement PARAM, PARAMCD, and PARAMN.

By intuition, many programmers will simply implement PARCAT1 as a copy of XOSCAT, PARAM as a copy of XOTEST, and PARCAT2 as a copy of XOLOC, which has the following implementation of ADXO:

Table 2. Incorrect Implementation of PARAM

SUBJID	USUBJID	PARCAT1	PARCAT2	PARCAT3	PARAM	AVAL	AVISIT
101-01	001-101-01	OS	Temporal	Study Eye	Fluorescein Comean Staining Score	5	Visit 2
101-01	001-101-01	OD	Temporal	Fellow Eye	Fluorescein Comean Staining Score	4	Visit 2
101-01	001-101-01	OS	Nasal	Study Eye	Fluorescein Comean Staining Score	3	Visit 2
101-01	001-101-01	OD	Nasal	Fellow Eye	Fluorescein Comean Staining Score	4	Visit 2
101-02	001-101-02	OS	Temporal	Fellow Eye	Fluorescein Comean Staining Score	3	Visit 2
101-02	001-101-02	OD	Temporal	Study Eye	Fluorescein Comean Staining Score	2	Visit 2
101-02	001-101-02	OS	Nasal	Fellow Eye	Fluorescein Comean Staining Score	4	Visit 2
101-02	001-101-02	OD	Nasal	Study Eye	Fluorescein Comean Staining Score	3	Visit 2

However, this implementation violates quite a few ADaM rules. First of all, PARAM in this implementation is not CDISC-compliant. According to ADaM IG Version 1.0, a parameter or PARAM need be defined in a way such that using a parameter one can unambiguously identify a record. PARAM does not have a qualifier. It is worth noticing that PARAM in ADaM is not always equivalent to --TEST in SDTM. In SDTM, some qualifier(s) can be added to --TEST to unambiguously identify a measurement. For instance, in SDTM, the definition of XOTEST is CDISC compliant since XOTEST plus qualifiers XOSCAT and XOLOC can be used to unambiguously identify a measurement. But ADaM requires that using a PARAM without any qualifiers one must be able to unambiguously identify a measurement. In the previous table, without PARCAT1 and PARCAT2, a PARAM cannot unambiguously identify the measurement.

Considering this, one may implement PARAM as follows:

Table 3. Implementation of PARAM

SUBJID	USUBJID	PARCAT1	PARAM	AVAL	AVISIT
101-01	001-101-01	Study Eye	Fluorescein Comean Staining Score (OS) Temporal	5	Visit 2
101-01	001-101-01	Fellow Eye	Fluorescein Comean Staining Score (OD) Temporal	4	Visit 2
101-01	001-101-01	Study Eye	Fluorescein Comean Staining Score (OS) Nasal	3	Visit 2
101-01	001-101-01	Fellow Eye	Fluorescein Comean Staining Score (OD) Nasal	4	Visit 2
101-02	001-101-02	Fellow Eye	Fluorescein Comean Staining Score (OS) Temporal	3	Visit 2
101-02	001-101-02	Study Eye	Fluorescein Comean Staining Score (OD) Temporal	2	Visit 2

101-02	001-101-02	Fellow Eye	Fluorescein Comean Staining Score (OS) Nasal	4	Visit 2
101-02	001-101-02	Study Eye	Fluorescein Comean Staining Score (OD) Nasal	3	Visit 2

Please note that the implementation of PARAM in Table 3 is CDISC-compliant. However, the implementation of PARCAT1 is not CDISC-compliant, yet. This will be explained later.

2.2 **Implementation of PARCATy**

PARCATy variables are used to categorize parameters, which implies that a PARCATy can includes one or more PARAM's and that a PARAM must belong to one PARCATy. In Table 3, Subject 001-101-01 has OS as the study eye, but Subject 001-101-02 has OD as the study eye. So when PARAM = "Fluorescein Comean Staining Score (OS) Temporal", it is the measurement for Study eye for Subject 001-101-01 and PARCAT1 = 'Study Eye'. However, PARCAT1 = 'Fellow Eye' for Subject 001-101-02. So in this case, the same PARAM belongs to different PARCAT1 depending on who the subject is. Therefore, whether this measurement is for Study eye or Fellow Eye cannot be implemented using variable PARCAT1. Instead, one may rename PARCAT1 to ACAT1, or create two flags, one to identify Study Eye, while the other to identify the Fellow Eye.

Table 4. Implementation of PARAM and PARCAT1

SUBJID	PARCAT1	ACAT1	PARAM	AVAL	AVISIT
101-01	OS	Study Eye	Fluorescein Comean Staining Score (OS) Temporal	5	Visit 2
101-01	OD	Fellow Eye	Fluorescein Comean Staining Score (OD) Temporal	4	Visit 2
101-01	OS	Study Eye	Fluorescein Comean Staining Score (OS) Nasal	3	Visit 2
101-01	OD	Fellow Eye	Fluorescein Comean Staining Score (OD) Nasal	4	Visit 2
101-02	OS	Fellow Eye	Fluorescein Comean Staining Score (OS) Temporal	3	Visit 2
101-02	OD	Study Eye	Fluorescein Comean Staining Score (OD) Temporal	2	Visit 2
101-02	OS	Fellow Eye	Fluorescein Comean Staining Score (OS) Nasal	4	Visit 2
101-02	OD	Study Eye	Fluorescein Comean Staining Score (OD) Nasal	3	Visit 2

It is worth pointing out that this CDISC-compliant implementation is a little awkward for table programming. For example, analyzing the FCS scores on nasal side involves two different PARAM's, depending on which eye is the Study Eye.

2.3 Implementation of by-visit Baseline

For ophthalmology trials, it is quite often that the difference of pre-dose and post-dose intraocular pressures (IOP) need be analyzed. In this case, we can define the pre-dose intraocular pressure at each visit as the baseline for the records measured at each visit. So the implementation will be as follows:

Table 5. Implementation of by-visit baseline

SUBJID	PARAM	AVISIT	ATPT	ABLFL	AVAL	BASE	CHG	BASETYPE
101-01	IOP (mmHg) (OD)	Visit 2	Predose	Y	20	20	0	Baseline for Visit 2
101-01	IOP (mmHg) (OD)	Visit 2	Postdose		30	20	10	Baseline for Visit 2
101-01	IOP (mmHg) (OD)	Visit 3	Predose	Y	21	21	0	Baseline for Visit 3
101-01	IOP (mmHg) (OD)	Visit 3	Postdose		19	21	-2	Baseline for Visit 3
101-01	IOP (mmHg) (OD)	Visit 4	Predose	Y	22	22	0	Baseline for Visit 4
101-01	IOP (mmHg) (OD)	Visit 4	Postdose		25	22	3	Baseline for Visit 4

In this table, BASETYPE is needed since more than one record within one PARAM is flagged by ABLFL. According to ADaM IG V1.0, if there are multiple baseline records flagged for a given parameter within a subject then BASETYPE should be populated and contain different values for the baseline records within a subject. Please note that BASETYPE cannot be simply set as "By-visit Baseline".

2.4 Implementation of by-time-point Baseline

Similarly, in some single period studies, analyses need be performed at multiple time points. For instance, in ophthalmology studies, we may need to compare the effect of a study drug and a competitor over a 24-hour period. In order to do this, the intraocular pressure is measured at 12 planned time points in 24 hours. To facilitate the analyses, a baseline can be defined for each planned time point.

Table 6. Implementation of by-time-point baselines

SUBJID	**PARAM**	**AVISIT**	**ATPT**	**ABLFL**	**AVAL**	**BASE**	**CHG**	**BASETYPE**
1001	IOP (mmHg) (OD)	Visit 2	8 AM	Y	16	16	0	Baseline at 8 AM
1001	IOP (mmHg) (OD)	Visit 2	10 AM	Y	17	17	0	Baseline at 10 AM

1001	IOP (mmHg) (OD)	Visit 2	12 PM	Y	15	15	0	Baseline at 12 PM
1001	IOP (mmHg) (OD)	Visit 2	2 PM	Y	19	19	0	Baseline at 2 PM
1001	IOP (mmHg) (OD)	Visit 2	4 PM	Y	20	20	0	Baseline at 4 PM
1001	IOP (mmHg) (OD)	Visit 3	8 AM		18	16	2	Baseline at 8 AM
1001	IOP (mmHg) (OD)	Visit 3	10 AM		21	17	4	Baseline at 10 AM
1001	IOP (mmHg) (OD)	Visit 3	12 PM		17	15	3	Baseline at 12 PM
1001	IOP (mmHg) (OD)	Visit 3	2 PM		14	19	-5	Baseline at 2 PM
1001	IOP (mmHg) (OD)	Visit 3	4 PM		20	20	0	Baseline at 4 PM

In this example, a baseline is defined for each planned time point. As a result, BASETYPE has a different value for each time point.

2.5 Implementation of multiple baselines

For multi-period studies, it may be needed to define multiple baselines for the same measurements for different analyses. For example, for a two-period study, there are two definitions of baselines: one is the last value in screening period and the other is the last value in Period 1 or wash-out period. For this type of study design, there are two different situations:

1) For records in the first period, baseline is from screening period; for Period 2 records, two different baselines are both defined - one is the last record in the screening period and the other one is the last record in Period 1 or wash-out period.

Given that, in ADaM standards, variable BASE is the only variable to store baseline value (which implies that one cannot add a variable such as BASE2 to store the second baseline), one need create a new record for each Period 2 record and the Period 1 record that is the baseline for Period 2. In this case, we have two sets of records and BASETYPE is used to distinguish these two sets of records.

Table 7. Implementation of multi-baselines for multi-period studies

SRCSEQ	SUBJID	PARAM	AVISIT	APHASE	ABLFL	AVAL	BASE	CHG	BASETYPE
1	1001	ALT (U/L)	Visit 1	Screening		22	20	.	SCREENING
2	1001	ALT (U/L)	Visit 2	Screening	Y	20	20	0	SCREENING
3	1001	ALT (U/L)	Visit 3	Period 1		30	20	10	SCREENING
4	1001	ALT (U/L)	Visit 4	Period 1		32	20	12	SCREENING

5	1001	ALT (U/L)	Visit 5	Period 2		21	20	1	SCREENING
6	1001	ALT (U/L)	Visit 6	Period 2		25	20	5	SCREENING
4	1001	ALT (U/L)	Visit 4	Period 1	Y	32	32	0	PERIOD 1
5	1001	ALT (U/L)	Visit 5	Period 2		21	32	-11	PERIOD 1
6	1001	ALT (U/L)	Visit 6	Period 2		25	32	-7	PERIOD 1

In the cases of by visit or by time point baselines (Tables 5 and 6), there are multiple baseline definitions, but only one baseline is defined for any single record. As a result, there is no need to create any new records. However, in the case of multi-baselines in a multi-period study, Period 2 records have two baselines defined. So a new set of records are created. From these examples, one can see that whether to create new sets of records depends on whether multiple baselines are defined for a single record.

2) For records in the first period, baseline is from screening period; for Period 2 records, the baseline is the last record in Period 1.

Table 8. Implementation of multi-baselines for a multi-period study

SRCSEQ	SUBJID	PARAM	AVISIT	APHASE	ABLFL	AVAL	BASE	CHG	BASETYPE
1	1001	ALT (U/L)	Visit 1	Screening		22	20	.	SCREENING
2	1001	ALT (U/L)	Visit 2	Screening	Y	20	20	0	SCREENING
3	1001	ALT (U/L)	Visit 3	Period 1		30	20	10	SCREENING
4	1001	ALT (U/L)	Visit 4	Period 1		32	20	12	SCREENING
4	1001	ALT (U/L)	Visit 4	Period 1	Y	32	32	0	PERIOD 1
5	1001	ALT (U/L)	Visit 5	Period 2		21	32	-11	PERIOD 1
6	1001	ALT (U/L)	Visit 6	Period 2		25	32	-7	PERIOD 1

In this implementation, Visit 4 record will be analyzed the same way as the other Period 1 records. In addition, Visit 4 record is also the baseline for Period 2 records. A copy of this record had better be added and the new record is flagged as the baseline for Period 2 records.

2.6 Implementation of lab data in standard and conventional units

It is not rare that in lab dataset ADLB, results in both standard unit and conventional units need be analyzed. In some cases, adjusted measurements may be analyzed. Therefore, in principle, for each lab test, it is better to create a PARAM and PARAMCD for standard unit and conventional unit, respectively. Intuitively, one may use PARCATy to distinguish standard unit and

conventional unit, while use lab test plus the standard unit and conventional unit, respectively, to create PARAM. So one may implement ADLB as follows:

Table 9. Implementation of lab data in standard unit and conventional units

SRCSEQ	SUBJID	PARCAT1	PARAM	PARAMCD	AVISIT	AVAL
1	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 1	22
2	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 2	20
3	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 3	30
4	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 4	32
1	1001	Lab in conventional unit	ALT (g/L)	ALT	Visit 1	xx
2	1001	Lab in conventional unit	ALT (g/L)	ALT	Visit 2	xx
3	1001	Lab in conventional unit	ALT (g/L)	ALT	Visit 3	xx
4	1001	Lab in conventional unit	ALT (g/L)	ALT	Visit 4	xx

Keep in mind that the standard unit and conventional units are different for most of the lab tests, while they are identical for some tests. This makes the creation of PARAM and PARAMCD more complex, given that PARAMCD and PARAM must be one-to-one correspondence. In the previous implementation, PARAM and PARAMCD are not one-to-one. So the ADaM-compliant implementation can be as follows:

Table 10. Implementation of lab data in standard unit and conventional units

SRCSEQ	SUBJID	PARCAT1	PARAM	PARAMCD	AVISIT	AVAL
1	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 1	22
2	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 2	20
3	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 3	30
4	1001	Lab in standard unit	ALT (mol/L)	ALT	Visit 4	32
1	1001	Lab in conventional unit	ALT (g/L)	ALTC	Visit 1	xx
2	1001	Lab in conventional unit	ALT (g/L)	ALTC	Visit 2	xx
3	1001	Lab in conventional unit	ALT (g/L)	ALTC	Visit 3	xx
4	1001	Lab in conventional unit	ALT (g/L)	ALTC	Visit 4	xx

2.7 Implementation of LOCF when there are 'NOT DONE' records

In SDTM findings domains, there can be records with --STAT = 'NOT DONE'. For example, QS domain can be like Table 11, in which Subject 1001 has QSSTAT = 'NOT DONE' at Visit 3.

Table 11. Questionnaire with Missing Values in SDTM QS Domain

SUBJID	QSTEST	VISIT	QSSEQ	QSSTRESN	QSSTAT
1001	Question #1	Visit 1	1	24	
1001	Question #1	Visit 2	2	25	
1001	Question #1	Visit 3	3		NOT DONE
1001	Question #1	Visit 4	4	28	

In ADaM datasets, last observation carried forward (LOCF) approach is often used to impute the missing values. For the above example, Visit 2 value need be carried forward to Visit 3. Basically, there are two possible CDISC-compliant implementation of LOCF when there are some source records in SDTM with --STAT = 'NOT DONE'. The first one is to utilize these records with --STAT = 'NOT DONE' and copy AVAL from last records.

Table 12. Implementation of LOCF - approach #1

SUBJID	PARAM	AVISIT	VISIT	AVAL	DTYPE	SRCSEQ	QSSEQ	QSSTRESN	QSSTAT	ANL01FL
1001	Question #1	Visit 1	Visit 1	24		1	1	24		Y
1001	Question #1	Visit 2	Visit 2	25		2	2	25		Y
1001	Question #1	Visit 3	Visit 3	25	LOCF	2	3	.	NOT DONE	Y
1001	Question #1	Visit 4	Visit 4	28		4	4	28		Y

Note that this approach does not create a new record for Visit 3. Instead, only the value at Visit 2 is copied to Visit 3. To keep traceability, QSSEQ is kept and is different from SRCSEQ.

There can be a second approach to the implementation of LOCF. In this approach, all the records with --STAT = 'NOT DONE' will be ignored and a new record for each visit will be created. For the newly created records, DTYPE = 'LOCF'.

Table 13. Alternative Implementation of LOCF - approach #2

SUBJID	PARAM	AVISIT	VISIT	AVAL	DTYPE	SRCSEQ	QSSEQ	QSSTRESN	QSSTAT	ANL01FL
1001	Question #1	Visit 1	Visit 1	24		1	1	24		Y
1001	Question #1	Visit 2	Visit 2	25		2	2	25		Y
1001	Question #1	Visit 3	Visit 3			3	3	.	NOT DONE	
1001	Question #1	Visit 3		25	LOCF	2		.		Y
1001	Question #1	Visit 4	Visit 4	28		4	4	28		Y

In this approach, the 'NOT DONE' record (record #2) is kept and ANL01FL = null. A new record (Record #4) is created whose AVAL is copied from the second record. In the new record, SRCSEQ = 2 while QSSEQ is missing.

The first approach results in less records but makes changes to the records with --STAT = 'NOT DONE'. The second approach does not make changes to any records from the source data and some new records are created. Both approaches are CDISC-compliant and correctly implement LOCF method. It is the user's preference to choose the approach.

CONCLUSIONS

This paper presented the author's thoughts and practice on several important issues about implementing CDISC-compliant ADaM datasets for ophthalmology studies. Making good choices on these issues can help make the datasets CDISC-compliant and serve the analysis purpose well.

REFERENCES

[1] CDISC Analysis Data Model ,Version 2.1.

[2] CDISC ADaM Implementation Guide, Version 1.0.

[3] CDISC SDTM Implementation Guide, Version 3.1.2.

[4] Nate Freimark, Susan Kenny, Jack Shostak, and John Troxell. Common Misunderstandings about ADaM Implementation. PharrnaSug 2012. San Francisco, CA. May 8 - 11, 2012.

[5] Songhui Zhu. Designing and Tuning ADaM Datasets. PharmaSUG 2013, Chicago, IL. May 13 - 15, 2013.

[6] Songhui Zhu and Lin Yan. Methods of Building Traceability for ADaM Data. ParmaSUG 2011, Nashville, TN. May 9 – 11, 2011.

CONTACT INFORMATION

Songhui Zhu, PhD

A2Z Scientific Inc.

East Brunswick, NJ 08816

songhui.zhu@a2zscientific.com