A Complete Derivation of Duration and Display in ISO 8601 Using SAS Program

Joyce Gui, Helen Wang Merck & Company
Sandy Wang Rutgers, The State University of New Jersey

Abstract

In clinical trials, an adverse event (AE) can occur anytime during the course of a study. In data analysis, it is common to relate AE information with study medication which is referred to as *time since last dose*. If the duration variable is included for regulatory submission, it must follow the ISO 8601 duration formats as described in the CDISC SDTM Implementation Guide (Version 3.1.1) section 4.1.4.3. This paper will use the adverse experience start date/time and the study medication date/time to derive the time since last dose based on defined rules using SAS® programs, and display the duration in ISO 8601 duration format.

Introduction

Time since last dose (referred to as variable AEDOSDUR) should be calculated by comparing the AE start date/time to the study medication date/time, where the number of doses is greater than zero, and is immediately prior or equal to the AE start date/time. This paper will provide the complete set of rules and duration display format in a flow chart for AEDOSDUR when the adverse event and study medication date and/or time are collected. As an example, some SAS programs and macros are included to demonstrate the derivations.

Rules and Definitions

In general, when time is collected and included in both the AE Start Date/Time and Study Medication Date/Time it will be used in the calculation. Derivation results will be in hours, minutes, and/or seconds, or they will include days and time of duration if it is greater than 24 hours. If time is collected and included in only one of, or in neither of, the input dates, then the derivation results will be specific to *days since last dose* [1].

Per CDISC SDTM IG 3.1.1 section 4.1.4.3, when duration is being measured *after* an event, the duration is displayed as: YYYY-MM-DDThh:mm:ss /PnYnMnDTnHnMnS. For duration measured *prior to* an event, the syntax is: PnYnMnDTnHnMnS/YYYY-MM-DDThh:mm:ss. If duration = 0, then it will be displayed as YYYY-MM-DD/PT0M. [2]

The following variables are defined and used in the flow chart to demonstrate the rules and derivation logic:

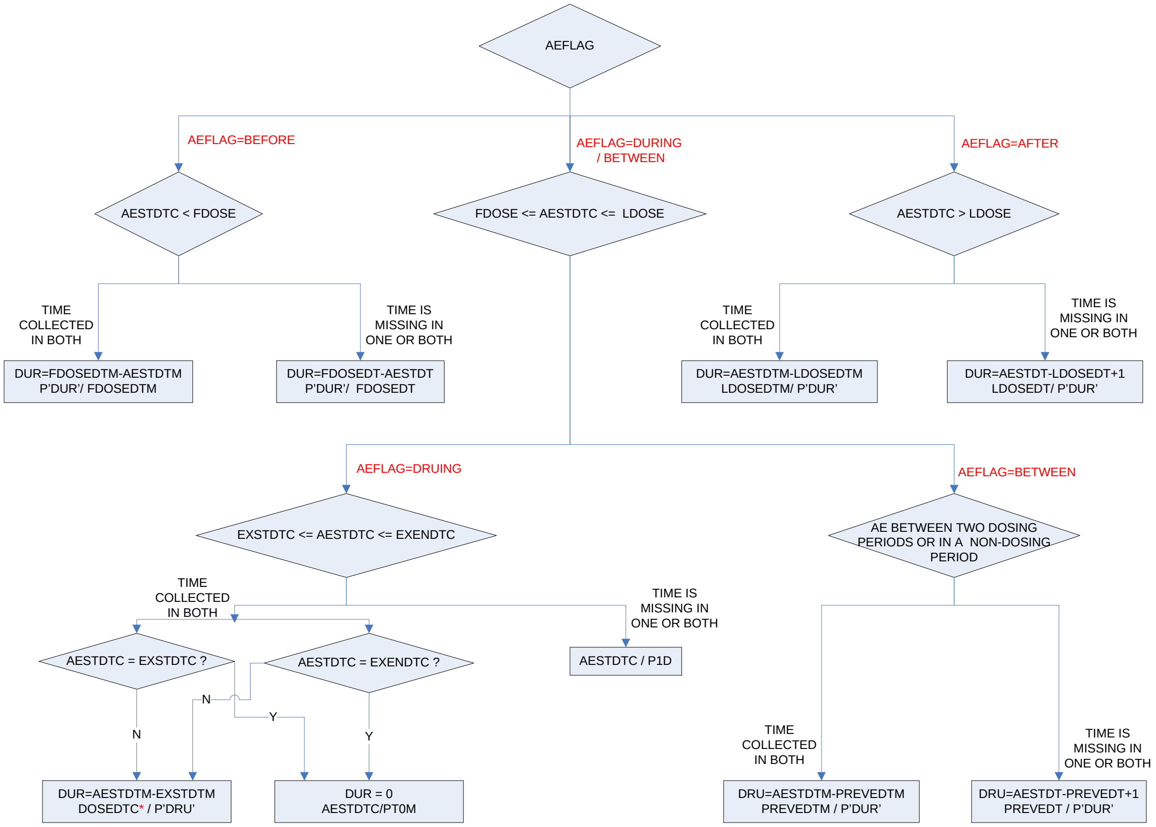
AEFLAG is a variable to be used as a flag to identify the occurrence of an AE start date/time (AESTDTC) relative to the study medication date/time (EXSTDTC,

EXENDTC) where EXSTDTC is the start date/time and EXENDTC is the end date/time of the study medication (SM) dosing period.
AEFLAG =

- BEFORE means the AE starts before the first SM taken (i.e. AESTDTC < FDOSE) where FDOSE is the first SM dose date/time.
- AFTER means AE starts after last SM taken (i.e. AESTDTC > LDOSE) where LDOSE is the last SM dose date/time.
- BETWEEN means AE starts between two dosing periods (or in a non-dosing period) and the AE start date/time is not equal to any of the SM date/time (i.e. AESTDTC > PREVDOSE and AESTDTC < NEXTDOSE) where PREVDOSE is the EXENDTC of previous dosing period and NEXTDOSE is the EXSTDTC of next dosing period.
- DURING means AE starts during a continuous dosing period (i.e. EXSTDTC* <=AESTDTC <=EXENDTC).

** When date part of AESTDTC is greater than date part of EXSTDTC, the EXSTDTC will be the DOSEDTC, which is the study medication start date immediately prior or equal to the AE start date.*

Flow Chart


AEFLAG
AEFLAG=BEFORE
AEFLAG=DURING / BETWEEN
AEFLAG=AFTER
AESTDTC < FDOSE
FDOSE <= AESTDTC <= LDOSE
AESTDTC > LDOSE
TIME COLLECTED IN BOTH
TIME IS MISSING IN ONE OR BOTH
DUR=FDOSEDTM-AESTDTM P'DUR'/ FDOSEDTM
DUR=FDOSEDT-AESTDT P'DUR'/ FDOSEDT
TIME COLLECTED IN BOTH
TIME IS MISSING IN ONE OR BOTH
DUR=AESTDTM-LDOSEDTM LDOSEDTM/ P'DUR'
DUR=AESTDT-LDOSEDT+1 LDOSEDT/ P'DUR'
AEFLAG=DRUING
AEFLAG=BETWEEN
EXSTDTC <= AESTDTC <= EXENDTC
AE BETWEEN TWO DOSING PERIODS OR IN A NON-DOSING PERIOD
TIME COLLECTED IN BOTH
TIME IS MISSING IN ONE OR BOTH
AESTDTC = EXSTDTC ?
AESTDTC = EXENDTC ?
AESTDTC / P1D
N
N
Y
Y
DUR=AESTDTM-EXSTDTM DOSEDTC* / P'DRU'
DUR = 0 AESTDTC/PT0M
TIME COLLECTED IN BOTH
TIME IS MISSING IN ONE OR BOTH
DRU=AESTDTM-PREVEDTM PREVEDTM / P'DUR'
DRU=AESTDT-PREVEDT+1 PREVEDT / P'DUR'

Example Datasets

Two input datasets AE, EX and an output dataset AEDUR are given here to illustrate the derivations.

Input data AE:

USUBJID	AESEQ	AETERM	AESTDTC	AEENDTC
999_00001	1	Gastritis	2008-04-30T04:03	
999_00002	1	Hypertension	2008-03-23T09:00	
999_00003	1	Headache	2008-03-17T08:28	
999_00003	2	Ear infection	2008-03-27T09:00	
999_00004	1	Headache	2008-04-17T05:05	
999_00004	2	Ear infection	2008-04-19T04:06	
999_00005	1	Hypertension	2008-02-25T09:15	2008-02-26T10:21
999_00006	1	Headache	2008-03-03T12:05	
999_00007	1	Asthma	2008-06-07T12:04	
999_00008	1	Gastritis	2008-04-05T05:05	2008-03-03
999_00008	2	Gastritis	2008-05-25T06:50	2008-05-27T08:15

Input data EX:

USUBJID	EXSTDTC	EXENDTC	EXNUMDOS
999_00001	2008-03-28T05:04	2008-04-10	4
999_00001	2008-04-12T04:05	2008-04-24	2
999_00001	2008-04-25T03:01	2008-05-01	2
999_00002	2008-03-23T08:10	2008-03-23	2
999_00003	2008-02-25T07:18	2008-03-09	4
999_00003	2008-03-11T09:00	2008-03-19	2
999_00004	2008-04-11T07:11	2008-04-19	4
999_00004	2008-04-11T09:02	2008-04-19	4
999_00005	2008-03-04T06:23	2008-03-17	2
999_00005	2008-03-18T04:04	2008-03-31	2
999_00006	2008-01-25T10:00	2008-01-25	1
999_00007	2008-02-25T12:00	2008-02-25	1
999_00008	2008-03-25T19:00	2008-03-25	2
999_00008	2008-04-25T21:00	2008-04-25	1

Output data AEDUR:

USUBJID	AESEQ	AETERM	AESTDTC	DOSEDTC	AEFLAG	AEDOSDUR
999_00001	1	Gastritis	2008-04-30T04:03	2008-04-30	during	2008-04-30/P1D
999_00002	1	Hypertension	2008-03-23T09:00	2008-03-23T08:10	after	2008-03-23T08:10/PT50M
999_00003	1	Headache	2008-03-17T08:28	2008-03-17	during	2008-03-17/P1D
999_00003	2	Ear infection	2008-03-27T09:00	2008-03-19	after	2008-03-19/P9D
999_00004	1	Headache	2008-04-17T05:05	2008-04-17	during	2008-04-17/P1D
999_00004	2	Ear infection	2008-04-19T04:06	2008-04-19	after	2008-04-19/P1D
999_00005	1	Hypertension	2008-02-25T09:15	2008-03-04T06:23	before	P7DT21H8M/2008-03-04T06:23
999_00006	1	Headache	2008-03-03T12:05	2008-01-25T10:00	after	2008-01-25T10:00/P1M7DT2H5M
999_00007	1	Asthma	2008-06-07T12:04	2008-02-25T12:00	after	2008-02-25T12:00/P3M13DT4M
999_00008	1	Gastritis	2008-04-05T05:05	2008-03-25T19:00	between	2008-03-25T19:00/P10DT10H5M
999_00008	2	Gastritis	2008-05-25T06:50	2008-04-25T21:00	after	2008-04-25T21:00/P29DT9H50M

Conclusion

This paper provides a complete derivation of duration when event date and/or time present. Display the duration in ISO 8601 duration format is required if this variable is included in the SUPPQUAL for submission. The macros in Appendix are flexible to be used in SAS data steps for the need of calculation duration between two events, display the duration in ISO 8601 format.

Appendix

1. Macro Display_DurIS8601_Time

This macro assembles the calculated year, month, day, hour, minute and second, displays them in ISO 8601 format - this macro works in a SAS data step.

```
%macro Display_DurIS8601_Time (yeardur, mondur, daydur, hrdur, mindur, secdur, dur);
    if &yeardur > 0 then nY = put(&yeardur, best.)||'Y';
    if &mondur > 0 then nM = put(&mondur, best.)||'M';
    if &daydur > 0 then nD = put(&daydur, best.)||'D';

    if &hrdur > 0 then nH = put(&hrdur, best.)||'H';
    if &mindur > 0 then nMI = put(&mindur, best.)||'M';
    if &secdur > 0 then nS = put(&secdur, best.)||'S';

    length &dur $20.;

    if &hrdur > 0 or &mindur > 0 or &secdur > 0 then
        &dur = compress(nY||nM||nD||'T'||nH||nMI||nS);
    else
        &dur = compress(nY||nM||nD);

    drop nY nM nD nH nMI nS;
%mend;
```

2. Macro DurIS8601

This macro calculates calculates month, day and year duration between **sdate** and **edate**, calculates hour, minute and second duration between **stime** and **etime.** Parameter **sdate** is the start date and **edate** is the end date, **stime** is the start time and **etime** is the end time; **edate/etime** must be greater than or equal to **sdate/stime -** this macro works in a SAS data step.

```
%macro DurIS8601(sdate, edate, stime, etime);
    syear = year(&sdate);
    smonth = month(&sdate);
    sday = day(&sdate);
    eyear = year(&edate);
    emonth = month(&edate);
    eday = day(&edate);

    shour = hour(&stime);
    sminute = minute(&stime);
    ssecond = second(&stime);
    ehour = hour(&etime);
    eminute = minute(&etime);
    esecond = second(&etime);

    *** Calculate previous month/year of the &edate month ***;
    pmonth = emonth - 1;
    if pmonth = 0 then do;
        pmonth + 12;
        pyear = eyear - 1;
    end;
    else pyear = eyear;

    *** calculate month, day and year duration between &sdate and &edate ***;
    if emonth >= smonth then do;
        mondur = emonth - smonth;
        yeardur = eyear - syear;
    end;
    else if emonth < smonth then do;
        mondur = emonth - smonth + 12;
        yeardur = eyear - syear - 1;
    end;

    *** daydur = 0 is handled here ***;
    if eday >= sday then do;
        daydur = eday - sday;
        if daydur = 0 and timepart(&etime) < timepart(&stime) then do;
            daydur = intnx('month', mdy(pmonth, 1, pyear), 1) - intnx('month', mdy(pmonth,
1, pyear), 0);
            mondur = mondur - 1;
            if mondur = -1 and yeardur >=1 then do;
                yeardur = yeardur - 1;
                mondur = 11;
            end;
        end;
    end;
    else if eday < sday then do;
        daydur = (intnx('month', mdy(pmonth, 1, pyear), 1) - intnx('month', mdy(pmonth, 1,
pyear), 0)) + (eday - sday);
        mondur = mondur - 1;
        if mondur = -1 and yeardur >=1 then do;
            yeardur = yeardur - 1;
            mondur = 11;
        end;
    end;

    *** calculate hour, minute and second duration between &stime and &etime ***;
    if (timepart(&stime) > 0 and timepart(&etime) > 0) then do;
        if ehour >= shour then do;
            hrdur = ehour - shour;
        end;
        else if ehour < shour then do;
            hrdur = ehour - shour + 24;
            daydur = daydur - 1;
        end;

        if eminute >= sminute then do;
            mindur = eminute - sminute;
        end;
        else if eminute < sminute then do;
            mindur = eminute - sminute + 60;
            hrdur = hrdur - 1;
            if hrdur = -1 then do;
                hrdur = 23;
                daydur = daydur - 1;
            end;
        end;

        if esecond >= ssecond then do;
            secdur = esecond - ssecond;
        end;
        else if esecond < ssecond then do;
            secdur = esecond - ssecond + 60;
            mindur = mindur - 1;
            if mindur = -1 then do;
                mindur = 59;
                hrdur = hrdur - 1;
                if hrdur = -1 then do;
                    hrdur = 23;
                    daydur = daydur - 1;
                end;
            end;
        end;
    end;
%mend DurIS8601;
```

3. Program to create AEDOSDUR

After identifying the occurrence of an AE start date/time (AESTDTC) relative to the study medication date/time (EXSTDTC, EXENDTC) to get the values of AEFLAG, the AEDOSDUR can be derived by using the following program.

```
data aedur;
length aedosdur $50;
    set aeflag;
    if aeflag = 'during' then do;
        if datepart(aestdtm) = datepart(exstdtm) and index(aestdtc, 'T') > 0
            and index(exstdtc, 'T') > 0 then do;
            %DurIS8601(exstdt, aestdt, exstdtm, aestdtm);
            %Display_DurIS8601_t(yeardur, mondur, daydur, hrdur, mindur, secdur, dur);
            if dur = 'T' then dur = 'TOM';
            aedosdur = strip(exstdtc) ||'/P'||dur;
        end;
        else if aestdtm = exendtm and index(aestdtc, 'T') > 0
            and index(exendtc, 'T') > 0 then do;
            dur = 'TOM';
            aedosdur = strip(exendtc) ||'/P'||dur;
        end;
        else aedosdur = put(aestdt, is8601da.)||'/'||'P1D';
    end;

    else if aeflag = 'between' then do;
        %DurIS8601(prevedt, aestdt, prevedtm, aestdtm);

        if datepart(aestdtm) = datepart(exstdtm) and
            (index(aestdtc, 'T') = 0 or index(exstdtc, 'T') = 0) then do;
            aedosdur = put(aestdt, is8601da.)||'/P1D';
        end;

        else if (index(aestdtc, 'T') = 0 or index(prevedtc, 'T') = 0) then do;
            daydur = daydur + 1;
            %Display_DurIS8601(yeardur, mondur, daydur, dur);
            aedosdur = put(prevedt, is8601da.)||'/P'||dur;
        end;
        else if (index(aestdtc, 'T') > 0 and index(prevedtc, 'T') > 0) then do;
            %Display_DurIS8601_t(yeardur, mondur, daydur, hrdur, mindur, secdur, dur);
            aedosdur = strip(prevedtc) ||'/P'||dur;
        end;
    end;

    else if aeflag = 'before' then do;
        %DurIS8601(aestdt, fdosedt, aestdtm, fdosedtm);

        if (index(aestdtc, 'T') = 0 or index(fdosedtc, 'T') = 0) then do;
            %Display_DurIS8601(yeardur, mondur, daydur, dur);
            if dur = ' ' then do;
            dur = '1D';
            aedosdur = strip(aestdtc)||'/P'||dur;
            end;
            else do;
            aedosdur = 'P'||trim(dur)||'/'||put(fdosedt, is8601da.);
            end;
        end;
        else if (index(aestdtc, 'T') > 0 and index(fdosedtc, 'T') > 0) then do;
            %Display_DurIS8601_t(yeardur, mondur, daydur, hrdur, mindur, secdur, dur);
            aedosdur = 'P'||trim(dur)||'/'||strip(fdosedtc);
        end;
    end;

    else if aeflag = 'after' then do;
        %DurIS8601(ldosedt, aestdt, ldosedtm, aestdtm);

        if (index(aestdtc, 'T') = 0 or index(ldosedtc, 'T') = 0) then do;
            daydur = daydur + 1;
            %Display_DurIS8601(yeardur, mondur, daydur, dur);
            aedosdur = put(ldosedt, is8601da.)||'/P'||dur;
        end;
        else if (index(aestdtc, 'T') > 0 and index(ldosedtc, 'T') > 0) then do;
            %Display_DurIS8601_t(yeardur, mondur, daydur, hrdur, mindur, secdur, dur);
            aedosdur = strip(ldosedtc) ||'/P'||dur;
        end;
    end;
run;
```

Acknowledgement

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

References

[1] ShaoanYu & Joyce Gui SESUG 2008 BI-006
The Approach of Deriving Timing Variables in SDTM Standards
[2] CDISC SDTM Implementation Guide (Version 3.1.1)
[3] Merck & Co., Clinical Data Repository Standards Business Rules/Database Derivations version 4.2

Contact Information

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

Joyce Gui
Phone: 732-906-3183
Email: gui_joyce@yahoo.com

Helen Wang
Merck & Co
RY34-A320
216 E. Lincoln Ave
Rahway, New Jersey 07065
Phone: 732-594-1296
Email: helen_wang@merck.com

Sandy Wang
Phone: 609-947-2684
Email: zhi.s.wang@gmail.com