

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

### ABSTRACT

This paper discusses and lists a SAS® macro which addresses the challenge of evaluating timing variables and chronological sequencing for collected PK sampling and patient dosing data when sample and dose dates and times are not always recorded correctly on, or are omitted from, the case report form input. Examples of such datasets are PKS, POPPK, and NONMEM.

This macro is intended for use with the input data in CDISC format for a single dose per visit, with a single pre-dose sample and following post dose samples.

### INTRODUCTION

PK SAS data sets consist of two types of observation, *PK sampling*, and *dosing*. A 'PK sample' is a sample taken from the patient such as a blood draw or a urine specimen. A 'dose' is an administration of medication such as ingestion of pills or an IV infusion. A dose is usually preceded by one pre-dose sample and followed by one or more post-dose samples, usually at specified time intervals after the dose end time. In some studies there is more than one dose per day, such as an 'am dose' and a 'pm dose'. Both a PK sample and a dose are each referred to as an *event*. The chronological sequencing and time-points of both PK sampling and dosing are critical for accurate and meaningful analysis of PK data. Two types of PK timing are recorded, *nominal* time and *actual* time.

A nominal time is the named or scheduled time for a PK sample or a dose to take place printed on a case report form. Examples are "PREDOSE", "30 MINS POST DOSE".

An actual time is the time noted and recorded on the case report form by an investigator or other responsible healthcare professional drawing a PK sample or administering the medication. When medication is administered by IV infusion the infusion end time (or the infusion duration) is also recorded.

Whilst the vast majority of actual times are accurate, occasionally times are incorrectly recorded or are out of nominal time chronological sequence, such as a pre-dose PK sample after the dose time, or the date or time is incomplete or omitted.

This macro, named PKTIMES, addresses this issue and applies rules to create imputed dates and times.

### FUNCTIONAL DESCRIPTION

The PKTIMES macro performs the following functions:

1. Determines an event type as one of the following:
  - a) Pre-dose PK sample
  - b) Dose or infusion
  - c) Post-dose PK sample
2. Identifies missing, incomplete, or out of sequence dates and times
3. Imputes these dates and times following algorithms which make the best use of other related actual and nominal times.
4. Evaluates nominal and actual derived timing variables using the actual and imputed dates and times
5. Sets flags in observations found with inaccurate or questionable dates and times to indicate the nature of the inaccuracy and the method of imputation

### INPUT PK DATA

In CDISC PK data is normally collected as two files:

1. **PK** (Pharmacokinetic) – PK sample data (some studies may also include similar observations from a laboratory LB dataset)

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

2. **EX** (Exposure to study medication) – Dosing history

These two datasets must have been SET together as one combined SAS dataset with both the PC and EX observations, which must contain the following variables:

1. **Patient ID** (e.g. USUBJID, PATNUM). May be character or numeric but must be unique for each patient
2. **Event ID** (must be EVID). Must be a numeric integer set to 0 for each observation from PC (May be set to 2 for any observations which are to be excluded from the analysis but where time values are still required). Must be set to 1 for observations from EX.
3. **Sample Date/Time** from PC (e.g. PCDT). Must be character of length \$19 or greater. A correctly recorded date and time should be in SAS IS8601DT.datetime format using an upper case 'T' to separate the time from the date (e.g. "2014-06-02T14:23:00"). Leading zeroes must be present, but seconds can be omitted.
4. **Sample Nominal Time Point** from PC (e.g. PCTPT). Must be character and contain the case report form sample time text (e.g. "1 HOUR POST")
5. **Sample Study Day** from PC (e.g. PCDY). Must be numeric, integer, non-zero and non-missing
6. **Planned Visit Day** from PC and EX (VISITDY) must be numeric, integer, non-zero and non-missing
7. **Dose Start Date/Time** from EX (e.g. EXSTDTC) Must have the same attributes and format as PCDTC
8. **Dose End Date/Time** from EX (e.g. EXSTDTC) Must have the same attributes and format as PCDTC.
9. **Dose Study Day** from EX (e.g. EXSTDY) Must be numeric, integer, non-zero and non-missing

### INPUT PARAMETERS

The PKTIMES macro has the following input parameters. All of these parameters are mandatory but have the most common values as defaults:

1. **INDS**: Name of the input SAS dataset (PC and EX combined), this dataset is not changed by the macro
2. **OUTDS**: Name of the output dataset, created by the macro, this contains a copy of the input dataset plus the variables created by the macro (listed below) and sorted by patient and imputed actual time order
3. **PATNUM**: Patient ID variable (numeric or character), default is USUBJID
4. **PCTIME**: Variable from PC containing the specimen collection date and time, must be character and have a length of at least \$19 and with the date and time formatted as IS8601DT with a 'T' separating the date and the time, default is PCDTC
5. **PCTIMEPT**: Variable from PC containing the specimen nominal time point description or case report form text field from PC. Default is PCTPT
6. **PCTDAY**: Variable from PC containing the specimen collection numeric study day, default is PCDY
7. **DOSETIME**: Variable from EX containing the dose start date and time. Type and format must be similar to PCTIME. Default is EXSTDTC
8. **DOSEEND**: Variable from EX containing the infusion end date and time. Type and format must be similar to PCTIME. Default is EXENDTC
9. **DOSEDAY**: Variable from EX containing the numeric dose start study day, default is EXSTDY
10. **DEFDUR**: Default duration (numeric) in minutes, may be a hard coded number such as '60', or a column in the input dataset from EX containing the duration in minutes
11. **DEFTIME**: A default dose time hard coded in the format 'hh:mm:ss'. This value will only be used by the macro after all other imputation methods have been unsuccessfully attempted.
12. **NOMDAY**: Variable containing the nominal day of visit (numeric). Default is VISITDY. Should be only one per dose and associated PK sample draws
13. **NOMTCONU**: PK sampling nominal time units used in PC 'D'=Days (default), 'H'=Hours, or 'M'=Minutes
14. **NOMTDOSU**: Dosing nominal time units used in EX 'D'=Days (default), 'H'=Hours, or 'M'=Minutes

**A SAS® Macro to address PK timing variables issues**

Timothy J Harrington, Dataceutics inc, Pottstown, PA

15. **WINDOW:** User specified default time interval in minutes between a pre-dose sample and the corresponding dose start time
16. **TIMEUNIT:** Unit of time measurement of the derived timing variables added to the output dataset. 'D'=Days (default), 'H'=Hours, or 'M'=Minutes

*Note:* To reduce the amount of code and improve compile time performance there is no validation of the parameters within the macro code. The applicable SAS error or warning will be produced if a specified dataset cannot be accessed or a column is not found. Invalid or incorrectly formatted values may produce a syntax error or 'Invalid argument to input' run-time error.

**CREATED OUTPUT VALUES**

The following variables are added to the output dataset specified by the OUTDS parameter. Where no imputation is needed the actual date and time values are used, otherwise these are the imputed dates and times. Both nominal and actual times are evaluated in the units (days, hours, or minutes) specified by the TIMEUNIT parameter. Imputation flags indicate the reason for any imputation and the imputation method used (see below), these flags are left blank where no imputation is done. Holidays and Sundays are indicated with holiday flags. *Note:* Dates may be imputed and flags may be set because of values other than the dates being inconsistent with the dates, such as an incorrect visit day.

*Event Times*

EVNTDT Event date as a numeric SAS internal date formatted as DATE9.

EVNTSTTM Event time as a numeric SAS internal date formatted as TIME8.

EVNTSDTM Event date and time as a numeric SAS internal datetime formatted as DATETIME18.

EVNTDTC Character representation of the event date formatted as IS8601DA.

EVNTSTTC Character representation of the event time formatted as IS8601TM.

EVNTSDTC Character representation of the event date and time formatted as IS8601DT.

*Nominal Times*

NOMTMFDS Nominal Time from the Start of the First Dose

NOMTMMRD Nominal Time from the Start of the Most Recent Dose

NOMTMRDE Nominal Time from the End of the Most Recent Dose

*Actual Times*

ACTTMFDS Actual Time from the Start of the First Dose

ACTTMMRD Actual Time from the Start of the Most Recent Dose

ACTTMRDE Actual Time from the End of the Most Recent Dose

*Original Times*

ORGDATEC Character IS8601DA representation of the original date, including blank and incomplete dates.

ORGTIMEC Character IS8601DT representation of the original time, including blank and incomplete times, ':00' is appended for the seconds where only hours and minutes were specified

*Imputation Flags*

FLGDATE Flag indicating the reason for date imputation

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

FLGDATEI Flag indicating the date imputation method

FLGTIME Flag indicating the reason for time imputation

FLGTIMEI Flag indicating the time imputation method

### *Holiday Flags*

FLGOHOL Flag indicating an original date was a Sunday or a U.S. holiday

FLGHOL Flag indicating the imputed date is a Sunday or a U.S. holiday

## DATE IMPUTATION METHOD

Dates are imputed when they are incomplete such as '2014-06' or are out of sequence such as when one post-dose sample has a date of '2014-04-02' but all the other post-dose observations associated with the same dose are '2014-02-04'. The macro takes the most common date for the given study day and sets any missing, incomplete, or out of sequence dates to that common date. Where nominal sample times for a given dose extend past one day imputation is done by subtracting the visit day of the most recent sampling observation date from the current planned visit day plus one and then adding these days to the most recent date. A date (whether imputed or not) is incremented if the time imputation for the same observation results in a time going past midnight, or decremented if the imputed time rolls back into the prior day.

## PRE-DOSE SAMPLE TIME IMPUTATION METHOD

If a pre-dose sample time is missing, incomplete, or on or after the corresponding dose time that dose time less the user specified pre-dose window is used. If the dose time is missing or incomplete the post-dose times are checked in sequence from the earliest until a valid time is found. That post-dose time is then taken less its nominal time, default dose duration, and window time. If no usable post dose actual time is found the time of the prior pre-dose sample is used. If there is no prior pre-dose the imputed time is set to the user specified default time less the window time.

## DOSE OR INFUSION TIME IMPUTATION METHOD

If a dosing start time is missing, incomplete, or on or before the corresponding pre-dose sample time that pre-dose time plus the user specified pre-dose window is used. If the pre-dose time is missing or incomplete the post-dose times are checked in sequence from the earliest until a valid time is found. That post-dose time is then taken less its nominal time and the default duration. If no usable post dose actual time is found the time of the prior dose is used. If the prior dose does not have a usable time the prior pre-dose time plus the window time is used. If there is no prior dose the imputed time is set to the dose end time less the dose duration. If there is no dose end time or duration the user specified default time is used.

## POST-DOSE SAMPLE TIME IMPUTATION METHOD

If a post-dose time is missing, incomplete, on or before the dose time or out of sequence with other post-dose times imputation is performed. Unscheduled and Termination or Discontinuation visits are considered as post-dose. If the first post dose sample time needs to be imputed the dose end time (or start time plus duration) is used. Imputation of any subsequent post-dose samples takes the preceding post-dose sample and adds the nominal time difference. If the prior post dose does not have a usable time the following post-dose time is used after subtracting the nominal time difference. Otherwise the default dose time is taken plus the dose duration and that post-dose nominal time. Each post-dose sample is then taken in order from the earliest to the latest nominal time, the actual time sequence is checked against the prior post-dose time, whether actual or just imputed the time is re-imputed if it is on or earlier than the prior post-dose time so the correct post-dose sampling order is maintained.

## IMPUTATION FLAGS

The macro generates four imputation flags (listed above), these indicate the reason for imputation and the imputation method. Some studies have specific imputation rules which may differ from those used in this macro hence the flags can be used to identify where and why imputations have been performed. The following are the flag settings:

For dates, FLGDATE is set to 'M' for missing, 'I' for incomplete, or 'S' for out of sequence. FLGDATEI indicates the

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

imputation applied: 'C' for common date for the visit or 'V' for preceding visit date. If a date is rolled over midnight into the next day or the prior day because of the time imputation FLGDATEI is set to 'T' (tomorrow) or 'Y' (yesterday) accordingly. These flags are left blank where no imputation was attempted.

For times, FLGTIME is set to 'M' for missing, 'I' for incomplete, or 'S' for out of sequence. FLGDATEI indicates the imputation applied: 'D' used the dose start time, 'P' used the pre-dose time, 'E' used the dose end time (or start time plus duration), 'L' used the most recent valid dose time, 'Q' used the most recent valid pre-dose time, '1' through '9' indicates the post-dose nominal time used, 'F' used the user specified default time. These flags are left blank where no imputation was attempted

### HOLIDAY FLAGS

There are two holiday flags (listed above), which indicate a U.S. non-business day. FLGOHOL is for the original date and FLGIHOL is for the imputed date. Holidays are the six U.S. public holidays. An 'S' indicates a Sunday, 'N' New Year's Day (January 1), 'M' Memorial Day (Last Monday in May), 'I' Independence Day (July 4), 'L' Labor Day (First Monday in September), 'T' Thanksgiving Day (Thursday from Nov 22-29), and 'C' Christmas Day (December 25). When a holiday falls on a weekend the following Monday is flagged as that holiday, or when Independence Day or Christmas Day fall on a Saturday the preceding Friday is flagged as that holiday. Dates are not re-imputed because of Sundays or holidays. The holiday flags are left blank for all non-holiday weekdays and for Saturdays.

### OUTPUT DATA

The output dataset is named as the OUTDS parameter and is a copy of the input dataset with the output variables listed above added. This dataset is sorted by the imputed actual event datetimes (EVNTSDTM). There are no duplicates since duplicate observations in the input are removed and the imputation performed by the macro resolves time clashes. Where a date or time is missing or incomplete and there is no chronological information from which an imputed date and time can be derived the observation is dropped.

### ISSUES AND LIMITATIONS

The input dataset must conform to CDISC standards with dates and times as character strings formatted as SAS IS8601DT and contain the EVID flag.

The sampling time-points are hardcoded as the 'usual' values found on study case report forms. Unrecognized time-point texts may produce errors, or indicate a problem with a date or time but apply an inappropriate imputation. For post-dose visits, if a day and time are not specified in the time-point text the day and time are taken from the visit label.

Any duplicate observations are removed, or where there are multiple nominal samples of the same analyte at the same nominal time only the latest is kept, the reasoning is the earlier sample observations were being corrected by taking another sample. A similar rule is applied to multiple doses of the same medication at the same nominal time.

The regimen is assumed to be a pre-dose followed by a dose, followed by one or more post-dose samples. A pre-dose observation does not have to be present, neither does a post-dose. Unscheduled and termination observations may be present and are treated as post-dose, though where there is no meaningful nominal time indication imputation cannot be performed and that observation is dropped.

This macro handles only one medication at a time. If there are multiple medication types each type of dosing and only its associated PK sample observations should be handled separately by the macro in a given call.

Multiple analytes should be handled separately with their associated dosing observations.

This macro does not take into consideration infusion interruptions or missed doses (where one dose may involve two observations in the EX dataset). Issues such as this must be resolved before using this macro.

This macro is intended for use with all SAS versions 8 and 9 and has been tested on a UNIX platform. However, there are no calls to host services so it is likely to be compatible with all common operating systems.

### CONCLUSIONS

The PKTIMES macro has been designed to reduce programming and data analysis effort for both developers and QC programmers caused by incomplete and incorrectly sequenced dates and times and to resolve resulting ordering

**A SAS® Macro to address PK timing variables issues**

Timothy J Harrington, Dataceutics inc, Pottstown, PA

conflict which may occur between the nominal and actual event times. The sample and dose dates and times are critical for correct ordering of events and for evaluating how analyte concentration levels change over time in relation to the dosing. Valid dates and times are not changed by the macro, so if the sampling or dosing data is updated with corrected times the corrected values are kept when the macro is next run. For studies where there are specific rules for handling incomplete dates which are not part of the macro the imputation flags and the retained original dates and times may be used to identify 'problem' dates or times and form the basis of determining the appropriate corrective action.

**ACKNOWLEDGEMENTS**

Dataceutics inc., 1610 Medical Drive, Pottstown PA19464

Michael G. Wilson and Vincent Amoruccio, Section Co-Chairs Statistics and Pharmacokinetics, Pharmasug 2014

**CONTACT INFORMATION**

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

Timothy J. Harrington, Professional Services DataCeutics, Inc., 1610 Medical Dr, Ste 300, Pottstown, PA 19464

Office: 610 970 2333 Ext 8651

Email: harringt@dataceutics.com

Web: [www.dataceutics.com](http://www.dataceutics.com)

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.

**SOURCE CODE**

```

/*-----*
 * PKTIMES macro - imputes the dates and times of events and evaluates PK Nominal and Actual
 * timing variables. Input files are PC and EX set together and must be in CDISC format
 *-----*/
%macro pktimes(inds=pkt01, /* Input Dataset Name - this dataset is not changed by this macro */
               outds=d_pks, /* Output Dataset Name */
               patnum=usubjid, /* Patient ID (character or numeric) */
               pctime=pcdtc, /* Sample Date/Time 'YYYY-MM-DDTHH:MM:SS' (seconds optional)*/
               pctimept=pctpt, /* Sample timepoint from CRF such as PCTPT, SMPID. */
               pctday=pcdy, /* PK sample study day */
               dosetime=exstdtc, /* Dose Start Time as 'YYYY-MM-DDTHH:MM:SS' (seconds optional)*/
               doseend=exendtc, /* Dose End Time as 'YYYY-MM-DDTHH:MM:SS' (seconds optional) */
               doseday=exstdy, /* Dose or Infusion Start Study Day */
               defdur=60, /* Default (Nominal) Infusion Duration in Minutes */
               deftime=09:00:00, /* Default Dose Start Time as 'HH:MM:SS'(seconds optional) */
               nomday=visitdy, /* Nominal Study Day of Planned Event */
               nomtconu=D, /* PK Sampling Nominal Time Units (D=days, H=hours, M=minutes)*/
               nomtdosu=D, /* Dose Nominal Time Units (D=days, H=hours, M=minutes) */
               window=5, /* Nominal Time Window from Pre-Dose and Dose Start (minutes)*/
               timeunit=D); /* Derived Timing Variables Unit (D=days, H=hours, M=minutes) */

%local secinday;
%let secinday=86400;

/** Sort so all observations relating to the same nominal visit are grouped together **/

proc sort data=&inds out=inds01;
  by &patnum visitnum visit;
run;

/** Get nominal sample/dose event day and time from CRF labels, convert units as applicable **/
/** Nominal time is evaluated relative to the nominal dose time, hence the dose time is zero **/

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

data inds02(drop=fact);
  set inds01;
  by &patnum visitnum visit;
  length nomtname origdtm $25 orgdatec orgtimec $10;
  if evid in (0,2) then do;
    origdtm=left(uppercase(&pctime));
    refday=&nomday*( "%uppercase(%substr(&nomtconu,1,1))"='D')+
      ("%uppercase(%substr(&nomtconu,1,1))"='H')/24+
      ("%uppercase(%substr(&nomtconu,1,1))"='M')/1440);
    if refday=. then do; /* If nominal day is missing (e.g. unsheduled) use the actual day */
      refday=&pctday;
    end;
    if index(uppercase(&pctimept),'PRE')>0 then do;
      nomttime=-&window*60;
      nomtname='PRE DOSE';
    end;
    else if index(uppercase(&pctimept),'POST')>0 then do;
      fact=(index(uppercase(&pctimept),'DAY')>0)*&secinday+
        (index(uppercase(&pctimept),'HR')>0 or index(uppercase(&pctimept),'HOURL')>0 or
        index(uppercase(&pctimept),'H POST')>0)*3600+
        (index(uppercase(&pctimept),'MIN')>0)*60; /* Add any other CRF Post Dose texts here */
      if fact ne . then do;
        nomttime=input(scan(compress(uppercase(&pctimept),'ABCDEFGHIJKLMNOPQRSTUVWXYZ_-'),1,
          ' '),best.)*fact;
        end;
        nomtname='POST DOSE '||left(put(nomttime,8.));
      end;
      else do; /* Consider unscheduled, termination, and other visits as 'post dose' visits */
        nomttime=refday*&secinday;
        nomtname='POST DOSE '||trim(left(uppercase(visit)))||put(visitnum,8.);
      end;
    end;
  else if evid=1 then do;
    origdtm=left(uppercase(&dostime));
    refday=&doseday*( "%uppercase(%substr(&nomtdosu,1,1))"='D')+
      ("%uppercase(%substr(&nomtdosu,1,1))"='H')/24+
      ("%uppercase(%substr(&nomtdosu,1,1))"='M')/1440);
    nomttime=0;
    nomtname='DOSE START';
  end;
  if length(origdtm)>=10 then do; /* Keep the original CRF dates and times */
    orgdatec=substr(origdtm,1,10);
    orgtimec=substr(origdtm,12);
  end;
  else do;
    orgdatec=left(origdtm);
  end;
  if length(orgtimec)=5 then do; /* Add 00 seconds if only hours and minutes */
    orgtimec=trim(left(orgtimec))||':00';
  end;
  x=length(orgdatec); /* Complete date has a length of 10 (YYYY-MM-DD) */
  y=length(orgtimec); /* Complete time has a length of 8 (HH:MM:SS) */
  if length("&deftime")=5 then do; /* Set the default time as a SAS internal time */
    deftime=input(compress("&deftime"||':00',is8601tm.);
  end;
  else do;
    deftime=input(compress("&deftime",is8601tm.);
  end;
run;

/** Eliminate duplicates (including different assay observation from the same sample time) */
/** Eliminate duplicates where one observation is a missing or incomplete date and time and keep
the observation with the complete date and time */

proc sort data=inds02;
  by &patnum refday evid nomttime x y;

```

**A SAS® Macro to address PK timing variables issues**

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

run;

data inds03;
  set inds02;
  by &patnum refday evid nomttime x y;
  if last.nomttime;
run;

/** Arrange the dates and times in a horizontal format so the times from different observations
at the same visit can be compared ***/

proc sort data=inds03;
  by &patnum refday nomttime;
run;

data inds04;
  set inds03;
  by &patnum refday nomttime;
  length evntcol evntcolt $25;
  retain evntord 0;
  if first.refday then do;
    evntord=0;
  end;
  if nomtname='PRE DOSE' then do;
    evntcol='PR_';
  end;
  else if nomtname='DOSE START' then do;
    evntcol='DS_';
  end;
  else if index(nomtname,'POST') then do;
    evntcol='PD_';
    evntord=evntord+1;
    evntcol=trim(left(evntcol))||left(put(evntord,8.));
    evntcolt='PDT_'||left(put(evntord,8.));
  end;
  lastevnt=last.refday;
run;

proc transpose data=inds04(keep=&patnum refday evntcol orgtimec) out=hdt01(drop=_name_);
  by &patnum refday;
  id evntcol;
  var orgtimec;
run;

proc transpose data=inds04(keep=&patnum refday evntcolt nomttime where=(evntcolt ne ' '))
  out=hdt02(drop=_name_);
  by &patnum refday;
  id evntcolt;
  var nomttime;
run;

/** Get the most common date for the visit to use as an imputed date ***/

proc sort data=inds04(keep=&patnum refday orgdatec) out=com01;
  by &patnum refday orgdatec;
run;

data com02;
  set com01;
  by &patnum refday orgdatec;
  retain n 0;
  n=n*(first.orgdatec=0)+(length(orgdatec)=10);
run;

proc sort data=com02;
  by &patnum refday n;
run;

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

data com03(rename=(orgdatec=comdate));
  set com02;
  by &patnum refday n;
  if last.refday and n>0;
run;

/** Combine the horizontal formatted datasets and common dates dataset with original data **/

data inds05;
  merge inds04(in=a) hdt01(in=b) hdt02(in=c) com03(in=d)
        inds04(in=e keep=&patnum refday evntord lastevnt where=(lastevnt=1)
              rename=(evntord=evntcnt));
  by &patnum refday;
  if a;
run;

/** Get the maximum number of post dose events for any visit in the whole study **/

proc sql noprint;
  select distinct max(evntcnt) into :maxevnt from inds05;
quit;

/** Perform the date and time imputation, set the imputation flags **/

data inds06;
  set inds05;
  by &patnum refday;
  length lcomdate lpr_ lds_ $10 flgdate flgdatei flgtime flgtimei flgohol flgihol $1;
  retain lcomdate lpr_ lds_ ;
  retain limptime dosedur lrefday doseref 0;
  array posts {&maxevnt} pd_1 - pd_%left(&maxevnt); /* post dose actual times from dose start*/
  array nomtimes {&maxevnt} pdt_1 - pdt_%left(&maxevnt); /* post nominal times from dose strt*/
  format impdaten is8601da. imptimen limptime is8601tm.;
  if first.&patnum then do; /* initialize retained values */
    lpr_=' ';
    lds_=' ';
    dosedur=0;
    lcomdate=.;
    lrefday=.;
  end;
  if first.refday then do; /* Start of next visit */
    limptime=.;
  end;
  flgdate=scan('N M I S',1+(length(orgdatec)<=1)+2*(1<length(orgdatec)<10)+
              3*(length(orgdatec)=10 and orgdatec ne comdate)); /* Set bad date flag */
  if flgdate='N' then do;
    impdaten=input(orgdatec,is8601da.); /* Date is good */
  end;
  else do; /* Date is missing, incomplete, or */
    if length(comdate)=10 then do; /* out of sequence so use the */
      flgdatei='C'; /* common date for the visit */
      impdaten=input(comdate,is8601da.);
    end;
    else if length(lcomdate)=10 and refday ne . and lrefday ne . then do;
      flgdatei='V'; /* Otherwise use latest prior common date plus difference in visit days */
      impdaten=input(lcomdate,is8601da.)+(refday-lrefday);
    end;
  end;
  if nomtname='PRE DOSE' then do; /* Check the Predose time */
    doseref=refday;
    flgtime=scan('N M I',1+(length(pr_)<=1)+2*(1<length(pr_)<8));
    if length(pr_)=8 then do;
      imptimen=input(pr_,is8601tm.); /* Predose time is complete */
      if length(ds_)=8 then do; /* Predose time >= dose time, take dose time less window time */
        if input(pr_,is8601tm.)>=input(ds_,is8601tm.) then do;
          flgtime='S';
        end;
      end;
    end;
  end;

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

        flgtimei='D';
        imptimen=input(ds_,is8601tm.)-&window*60;
    end;
end;
else do;
    if length(ds_)=8 then do; /* Predose time incomplete, use dose time less window time */
        flgtimei='D';
        imptimen=input(ds_,is8601tm.)-&window*60;
    end;
    else do;
        do i=1 to evntcnt; /* Else try each Post Dose date, subtract its nominal time */
            if imptimen=. and length(posts{i})=8 and nomtimes{i} ne ' ' then do;
                flgtimei=compress(put(i,8.));
                imptimen=input(posts{i},is8601tm.)-nomtimes{i}-&defdur-&window*60;
            end;
        end;
    end;
    if imptimen=. then do; /* If no valid post times use predose time from the prior visit */
        if length(lpr_)=8 then do;
            flgtimei='Q';
            imptimen=input(lpr_,is8601tm.);
        end;
        else if length(lds_)=8 then do; /* Otherwise impute from dose time of prior dose */
            flgtimei='L';
            imptimen=input(lds_,is8601tm.)-&window*60;
        end;
        else do;
            flgtimei='F'; /* Otherwise use the default time */
            imptimen=deftime-&window*60;
        end;
    end;
end;
if &nomday=. then do;
    nomsecs=9999*&secinday; /* No nominal day, unscheduled, termination etc */
end;
else do;
    nomsecs=(&nomday-1)*&secinday-&window*60; /* Convert the nominal date to seconds */
end;
end;
else if nomname='DOSE START' then do; /* Check the dose or infusion start time */
    doseref=refday;
    flgtime=scan('N M I',1+(length(ds_)<=1)+2*(1<length(ds_)<8));
    if length(ds_)=8 then do;
        imptimen=input(ds_,is8601tm.); /* Dose start time is complete */
    end;
    else do;
        flgtimei='P'; /* If dose time is incomplete and Predose time ok use it plus window time */
        if length(pr_)=8 then do;
            imptimen=input(pr_,is8601tm.)+&window*60;
        end;
        else do;
            do i=1 to evntcnt; /* If no Predose time use next post time less its nominal time */
                if imptimen=. and length(posts{i})=8 and nomtimes{i} ne . then do;
                    flgtimei=compress(put(i,8.));
                    imptimen=input(posts{i},is8601tm.)-&defdur-datepart(nomtimes{i});
                end;
            end;
        end;
    end;
    if imptimen=. then do; /* Otherwise use the time of the prior dose */
        if length(lds_)=8 then do;
            flgtimei='L';
            imptimen=input(lds_,is8601tm.);
        end;
        else if length(lpr_)=8 then do; /* Else use time of prior Predose plus window */
            flgtimei='Q';
            imptimen=input(lpr_,is8601tm.)+&window*60;
        end;
    end;
end;

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

end;
end;
end;
if length(&doseend)=16 then do; /* Evaluate dose duration using the dose end time if */
  &doseend=trim(&doseend)||':00'; /* if available, otherwise use the given duration */
end;
if length(&doseend)=19 then do;
  dosendtm=input(substr(&doseend,12),is8601tm.);
end;
if dosendtm ne . and dosendtm>imptimen then do;
  dosedur=dosendtm-imptimen;
end;
else do;
  dosedur=&defdur*60;
end;
if imptimen=. then do;
  if dosendtm ne . then do; /* Use the dose end time */
    flgtimei='E';
    imptimen=dosendtm-dosedur;
  end;
  else do; /* Otherwise use the default time */
    flgtimei='F';
    imptimen=deftime;
  end;
end;
if &nomday=. then do;
  nomsecs=9999*&secinday; /* No nominal day, unscheduled, termination etc */
end;
else do;
  nomsecs=(&nomday-1)*&secinday; /* Convert the nominal dose date to seconds */
end;
end;
else if index(nomtname,'POST DOSE')>0 then do; /* Check Post Dose, Unscheduled etc */
  flgtime=scan('N M I',1+(length(posts{evntord})<=1)+2*(1<length(posts{evntord})<8));
  if length(posts{evntord})=8 then do;
    if input(posts{evntord},is8601tm.)<=limptime then do;
      flgtime='S'; /* Post dose time cannot be <= dose time or <= preceding post dose time */
      imptimen=limptime+nomtimes{evntord};
      if evntord=1 then do; /* Impute first post dose as dose end time + nominal time */
        flgtimei='E';
        imptimen=imptimen+dosedur;
      end;
    else if evntord>=2 then do; /* Impute subsequent post times by subtracting the prior */
      flgtimei=compress(put(evntord-1,8.)); /* post dose time */
      imptimen=imptimen-nomtimes{evntord-1};
    end;
  end;
  else do;
    imptimen=input(posts{evntord},is8601tm.); /* Post dose time good and in time sequence */
  end;
end;
else do; /* Post Dose time is missing or incomplete */
  if limptime ne . then do;
    imptimen=limptime+nomtimes{evntord}; /* Retain time of the prior post dose if valid */
    if evntord=1 then do;
      flgtimei='E';
      imptimen=imptimen+dosedur; /* Impute 1st post dose as dose end time + nominal time */
    end;
    else if evntord>=2 then do; /* Impute subsequent post times by subtracting the */
      flgtimei=compress(put(evntord-1,8.)); /* prior post dose time */
      imptimen=imptimen-nomtimes{evntord-1};
    end;
  end;
  else do; /* Otherwise use the next valid Post Dose time */
    if evntord<evntcnt then do;
      do i=evntord to evntcnt;
        if imptimen ne . and length(posts{i})=8 then do;

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

        flgtimei=compress(put(evntord-1,8.));
        imptimen=input(posts{i},is8601tm.)-(nomtimes{i}-nomtimes{evntord});
    end;
end;
end;
if imptimen=. then do; /* Otherwise use default dose time plus nominal Post Dose time */
    flgtimei='F';
    imptimen=deftime+dosedur+nomtimes{evntord};
end;
end;
end;
if &nomday=. then do;
    nomsecs=9999*&secinday; /* No nominal day, unscheduled, termination etc */
end;
else do;
    nomsecs=(doseref-1)*&secinday+nomtimes{evntord}; /* Convert nominal time to seconds */
end;
end;
if imptimen>=&secinday then do; /* If the imputed time rolls over 24 hrs increment the date */
    if index(nomname,'POST DOSE')=0 or flgdatei ne 'V' then do; /* for each 24 hrs, and the */
        impdaten=impdaten+int(imptimen/&secinday); /* remainder is the time */
        flgdate='S';
        flgdatei='T';
    end;
    imptimen=(imptimen/&secinday-int(imptimen/&secinday))*&secinday;
end;
if imptimen<0 then do; /* If imputed time rolls under midnight (<0) decrement date for each */
    if index(nomname,'POST DOSE')=0 or flgdatei ne 'V' then do; /*24 hrs, subtract remainder */
        impdaten=impdaten-(abs(int(imptimen/&secinday))+1); /* from 24hrs */
        flgdate='S';
        flgdatei='Y';
    end;
    imptimen=&secinday-(abs(imptimen));
end;
if flgdate='N' then do; /* Clear the flag when no imputation (original date or time is kept)*/
    flgdate=' ';
end;
if flgtime='N' then do;
    flgtime=' ';
end;
if length(orgdatec)=10 then do; /* Indicate if an original date was a holiday or a Sunday */
    w=mod(input(orgdatec,is8601da.),7);
    m=input(substr(orgdatec,6,2),8.);
    d=input(substr(orgdatec,9,2),8.);
    flgohol=scan('X N M I L T C',1+(m=1 and (d=1 or (w=3 and d in (2,3))))+
        2*(m=5 and w=3 and d>=25)+3*(m=7 and (d=4 or (w=3 and d=5) or (w=0 and d=3)))+
        4*(m=9 and w=3 and d<=7)+5*(m=11 and w=6 and d in (22,23,24,25,26,27,28))+
        6*(m=12 and (d=25 or (w=0 and d=24) or (w=3 and d=26))));
    if w=2 then do;
        flgohol='S';
    end;
    if flgohol='X' then do;
        flgohol=' ';
    end;
end;
if impdaten ne . then do; /* Indicate if an imputed date is a holiday or a Sunday */
    w=mod(impdaten,7);
    m=input(substr(put(impdaten,is8601da.),6,2),8.);
    d=input(substr(put(impdaten,is8601da.),9,2),8.);
    flgihol=scan('X N M I L T C',1+(m=1 and (d=1 or (w=3 and d in (2,3))))+
        2*(m=5 and w=3 and d>=25)+3*(m=7 and (d=4 or (w=3 and d=5) or (w=0 and d=3)))+
        4*(m=9 and w=3 and d<=7)+5*(m=11 and w=6 and d in (22,23,24,25,26,27,28))+
        6*(m=12 and (d=25 or (w=0 and d=24) or (w=3 and d=26))));
    if w=2 then do;
        flgihol='S';
    end;
    if flgihol='X' then do;

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

        flgihol=' ';
    end;
end;
output;
if length(comdate)=10 then do; /* Retain the date and nominal visit day to compare with a */
    lcomdate=comdate; /* date in the next visit if that date is missing */
    lrefday=refday;
end;
limptime=imptimen; /* Retain prior time value for possible use in the next */
if pr_ne ' ' then do; /* observation imputation in the same visit */
    lpr=pr_;
end;
if ds_ne ' ' then do;
    lds=ds_;
end;
run;

/** Use the imputed dates and times to get the date and time of first and subsequent doses */

data inds07(keep=&patnum impdaten imptimen nomsecs rename=(impdaten=datezero imptimen=timezero
nomsecs=nomtzero));
set inds06(where=(evid=1 and impdaten ne . and imptimen ne .)); /* Get the actual date and */
by &patnum; /* time of the first dose */
if first.&patnum;
run;

data inds08; /* Add the date and time of the first dose to every observation */
merge inds06(in=a) inds07(in=b);
by &patnum;
if a and b;
run;

proc sort data=inds08; /* Get the actual date and time for every dose for calculating time */
by &patnum doseref refday; /* from the most recent dose */
run;

data inds09(keep=&patnum doseref refday impdaten imptimen nomsecs rename=(impdaten=datedose
imptimen=timedose nomsecs=nomtdose));
set inds08(where=(evid=1 and impdaten ne . and imptimen ne .));
by &patnum doseref refday;
if first.refday;
run;

data inds10; /* Add the date and time of each dose to all observations with the same visit */
merge inds08(in=a) inds09(in=b);
by &patnum doseref;
if a and b;
run;

proc sort data=inds10;
by &patnum impdaten imptimen;
run;

/** Evaluate the Event times, the Nominal, Actual, and Imputed times ***/

data inds11;
set inds10;
by &patnum impdaten imptimen;
format evntsdm datetime20. evntdt date9. evntsttm time5.;
length evntsdtc $20 evntdtc $10 evntsttc $8;
retain nomtlast datelast timelast fact 0;
evntsdm=impdaten*&secinday+imptimen; /* SAS numeric datetime Event Date/Time */
evntdt=impdaten; /* SAS numeric date Event Date */
evntsttm=imptimen; /* SAS numeric time Event Time */
evntsdtc=left(put(evntsdm,is8601dt.));/* Character representation of event dates and times */
evntdtc=left(put(impdaten,is8601da.));
evntsttc=left(put(imptimen,is8601tm.));

```

## A SAS® Macro to address PK timing variables issues

Timothy J Harrington, Dataceutics inc, Pottstown, PA

```

if _n_=1 then do; /* Set a unit conversion factor based on the user defined timing units */
  fact=&secinday*( "%substr(%upcase(&timeunit),1,1)"='D')+
  3600*( "%substr(%upcase(&timeunit),1,1)"='H')+60*( "%substr(%upcase(&timeunit),1,1)"='M');
end;
if first.&patnum then do;
  nomtlast=.;
  datelast=.;
  timelast=.;
end;
if imptdaten=datezero and imptimen=timezero then do;
  nomtmfds=0; /* Nominal Time from First Dose */
  acttmfds=0; /* Actual Time from First Dose */
end;
else do;
  nomtmfds=(nomsecs-nomtzero)/fact;
  acttmfds=((imptdaten*&secinday+imptimen)-(datezero*&secinday+timezero))/fact;
end;
if evid=1 then do;
  nomtmrde=0; /* Nominal Time from Most Recent Dose */
  nomtmrde=0; /* Nominal Time from End of Most Recent Dose */
  nomtlast=nomtdose;
  acttmrde=0; /* Actual Time from Most Recent Dose */
  acttmrde=0; /* Actual Time from End of Most Recent Dose */
  datelast=datedose;
  timelast=timedose;
end;
else do;
  if nomtname='PRE DOSE' and timelast ne . then do;
    nomtmrde=(nomsecs-nomtlast)/fact;
    nomtmrde=(nomsecs-(nomtlast+&defdur*60))/fact;
    acttmrde=((imptdaten*&secinday+imptimen)-(datelast*&secinday+timelast))/fact;
    acttmrde=((imptdaten*&secinday+imptimen)-(datelast*&secinday+timelast+dosedur))/fact;
  end;
  else do;
    nomtmrde=(nomsecs-nomtdose)/fact;
    nomtmrde=(nomsecs-(nomtdose+dosedur))/fact;
    acttmrde=((imptdaten*&secinday+imptimen)-(datedose*&secinday+timedose))/fact;
    acttmrde=((imptdaten*&secinday+imptimen)-(datedose*&secinday+timedose+dosedur))/fact;
  end;
end;
run;

/** Drop temporary work variables, sort and create the output dataset **/

proc sort data=inds11 out=&outds(drop=comdate datelast datedose datezero deftime dosedur
  dosendtm ds_ evntcnt evntcol evntcolt evntord nomtdose nomtlast nomtname nomttime fact i
  imptdaten imptimen lastevnt lcomdate lds_ lrefday limptime lpr_ n nomsecs nomtdose nomtzero
  origdtm pdt_: pd_: pr_ refday timedose timelast timezero x y w m d);
  by &patnum evntsdtm;
run;

/** End of Macro and clean up **/

proc datasets nolist;
  delete inds: ;
quit;
run;

%mend pktimes;

```