MWSUG 2025 - Paper PL-018

Worried about that Second Date with ISO®? Using PROC FCMP to Convert and Impute ISO 8601 Dates to Numeric Dates

Richann Jean Watson, DataRich Consulting

ABSTRACT

Within the life sciences, programmers often find themselves doing a lot of dating – matching, converting between character and numeric values, and imputing missing components. Clinical Data Interchange Standards Consortium (CDISC) Study Data Tabulation Model (SDTM) domains have implemented the use of the International Organization for Standardization (ISO) format, ISO® 8601, for datetimes. These dates are stored as character strings with missing components denoted using a single hyphen. Although this format helps to standardize how dates and times are captured so that there is no confusion as to what the date represents, it leaves a longing for something more compatible for analysis purposes: determining durations and number of days from a reference point. The conversion of ISO dates to a numeric format requires a serious commitment, especially when partial ISO dates require imputations. Although SAS® offers a variety of built-in formats and functions that can appease both sides on a date, i.e., converting complete ISO dates to numeric values or numeric dates to ISO dates, there is no SAS-provided function that will help with the required conversion and imputation of a partial date. Fortunately, with the use of the FCMP procedure, we can create our own custom functions to help achieve our desired goal. This paper illustrates the process of building a custom function that will take a date that is captured in the appropriate ISO format in SDTM (--DTC) and convert that date to a numeric format while also giving partial dates the extra attention to impute missing components. Additionally, this custom function sets the correct date imputation variable (--DTF), so you always know just how much of a blind date your derived value really is.

INTRODUCTION

All dates within SDTM are stored as character strings following the ISO 8601 format. Character dates, however, are not conducive to performing common Analysis Data Model (ADaM) calculations, such as determining analysis day, duration of an event, assessing pre-treatment, on treatment/treatment emergence, or post-treatment. These calculations and date comparisons require the conversion of character ISO 8601 dates to numeric dates. Converting complete dates is straightforward using built-in SAS formats like E8601DA., but handling partial dates often requires additional processing, such as date imputations based on guidance from the Statistical Analysis Plan (SAP). This paper introduces a user-defined function, which allows either type of character date (i.e., the date that is easy and seamless or the date that needs that individual attention) to be converted to numeric dates. Plus, this user-defined function provides an indicator of the level of imputation applied, ensuring transparency in date handling

WHAT IS ISO 8601 FORMAT?

Before we begin constructing the user-defined function, it is essential to understand the ISO 8601 format. It is easy to misinterpret a date, especially with regional variations like mm/dd/yyyy or dd/mm/yyyy. ISO 8601 provides an internationally standardized way to represent dates and times: YYYY-MM-DD (ISO 8601 Date and Time Format, n.d.). For example, the internationally agreed way to represent the date July 9, 2021, is represented as 2021-07-09 in ISO 8601 format.

In addition, the format specification also instructs how a time component should be incorporated with a date to create a datetime. ISO 8601 orders components from highest date/time level to lowest date/time level—year, month, day, hour, minute, second—ensuring logical and chronological sorting. The standard format for a datetime is YYYY-MM-DDTHH:MM:SS, where:

- **YYYY** is the four-digit year
- **MM** *(first one in the format)* is the two-digit month, 01 - 12
- **DD** is the two-digit day, 01 - 28, 29, 30 or 31 depending on the month and year
- **T** separator and indicator that a time value follows

- **HH** is the two-digit hour, represented in 24-hour clock, 00 - 23
- **MM** *(second one in the format)* is the two-digit minutes, 00 - 59
- **SS** is the two-digit seconds, 00 - 59

ISO 8601 also specifies how to handle missing date (or time) components. Each missing component is represented by a single hyphen; and only components up to the last non-missing component are displayed. For example, a start date for a medication is required, but a patient does not recall the month they actually started taking the medication. The patient only recalls the year they started taking the medication and knows they take it on the 15th of every month. Therefore, the month is missing, and the estimated date would be represented as 2021--15 in the ISO 8601 format. A hyphen is used to represent the missing month.

SAMPLE DATA SET FOR PAPER

The adverse event onset datetimes and end datetimes in Data Display 1 need to be converted to a numeric date regardless of whether it is a complete or partial date. However, with partial dates, that extra attention to details needs to be considered to ensure the date conversion goes smoothly. With any real-world situation, the dates may vary in completeness. They can appear as full datetimes, full dates (without time), partial datetimes, partial dates, or missing. The goal is to extract and convert only the date portion to a numeric value.

AESTDTC	AEENDTC
2021	2021
2021-02	2021-03
2021-07-09	2022-09
2021-07-09T12:15	2021-09-09T12:15
2021-07	2022-08-09
2021	2022---09
2021-07--T12	2022
2021-07-09T12:15	
2021-07-09T12:15:45	2022-08--T12:15
2021-07-09T-:15	2022-08-09
2022	2022-02
2022-01	2024-02
	2022
2021---09T-:15	2022-08-09
--07-09	--08-09
-----T12:15	-----T12:15
----09	----09
2023	2023
2023-02	2023-03

Data Display 1: SDTM Dates

Note that AESTDTC and AEENDTC represent Adverse Event start and end dates in the SDTM AE domain; however, this could be applied to any time of data with partial dates.

PROC FCMP BASICS

As previously mentioned, SAS lacks a built-in format or function to convert partial character dates to numeric dates, so we have to create a user-defined function to properly impute and convert the dates in Data Display 1. The SAS Function Compiler procedure (PROC FCMP) allows users to define custom functions and subroutines. Although there are several options and statements that can be used in PROC FCMP, this paper only focuses on a small portion of the power of this procedure. The general syntax for creating custom functions or custom subroutines is shown in SAS Program 1.

```
PROC FCMP outlib = libname.dataset.package  <options>;
   SUBROUTINE subroutine-name (argument-1 <, argument-2, …>);
      OUTARGS out- argument-1 <, out-argument-2, …>;
      <<< SAS STATEMENTS >>>
   ENDSUB;
   FUNCTION function-name (argument-1 <, argument-2, …>) <$>;
      OUTARGS out- argument-1 <, out-argument-2, …>;
      <<< SAS STATEMENTS >>>
      RETURN(expression)
   ENDFUNC;
QUIT;
```

SAS Program 1: PROC FCMP General Syntax

The PROC FCMP statement has several options, **OUTLIB =**, which specifies a three-level name of an output data set where the compiled functions and subroutines are written once PROC FCMP hits a step boundary (i.e., QUIT statement), is required if you want to compile and save the user-defined function so that it can be subsequently called. An example of this option is **OUTLIB = FCMP.FUNCS.ISO_IMPDATE**. To make these custom functions and subroutines available within a DATA step, read about the CMPLIB= specification in the system OPTION statement as described in USING CUSTOM FUNCTIONS.

Defining a function or a subroutine must begin with a declaration statement (**FUNCTION** or **SUBROUTINE**, respectively) and conclude with a termination statement (**ENDFUNC** or **ENDSUB**).

Each function (or subroutine) requires a name and, for most of the functions (or subroutines), at least one argument (argument-1, argument-2). If the arguments are character, then they should have a dollar sign after the argument name; for example, in **ISOIMPDT(DATTIM $, REFDT, IMPUTFL $, IMPTYP $)**. DATTIM, IMPUTFL and IMPTYP are the three input character arguments, and therefore have the trailing $. REFDT is numeric; therefore, the $ is not needed.

Within PROC FCMP, the use of most DATA step statements is allowed. A function returns a value using the **RETURN** statement, and although multiple **RETURN** statements can exist within the same function, only one will execute during each function call. Therefore, only one value will be returned for each call to the function. If the return value is a character value, then the FUNCTION statement needs a dollar sign at the end, as seen in the example function **ISODTTM(DATTIM $) $**, as described in “Have a Date with ISO®? Using PROC FCMP to Convert Dates to ISO 8601” (Watson, 2022). Subroutines do not return a value and therefore the **RETURN** statement is not needed in the subroutine definition. However, both the FUNCTION and SUBROUTINE statements allow for multiple arguments to be modified within the function or subroutine. To modify arguments, the optional **OUTARGS** statement is used to indicate which arguments are to be changed in the calling program. (Hughes, SAS® Data-Driven Development: From Abstract Design to Dynamic Functionality, Second Edition, 2022)

CONVERT AND IMPUTE ISO 8601 DATES USING CUSTOM FUNCTION

When converting complete dates to a numeric date, SAS makes numeric dates a breeze with SAS-defined formats. However, when your date is not all there (i.e., containing missing components), special attention is needed for all the various imputation rules. The attentive companion typically relies on a tried-and-true technique, i.e. macros, to handle the various rules. The following user-defined function marries the best of SAS-defined functions and techniques normally done via macros. This function not only converts complete dates to numeric values but also imputes partial dates using customizable rules based on the arguments provided as well as setting the imputation indicator to the highest level of imputation.

In order to build a custom function robust enough to handle different imputation rules, several assumptions had to be made.

- Only dates will be imputed.
- Imputation occurs at the highest level of missingness (e.g., if year is provided and month is missing, then month is the highest level of missingness).
- A reference date (e.g., treatment start date) is used for some imputation rules.

- A different rule can be defined for handling partial dates that are clearly prior to the reference date.
- If more than one rule is specified in the function call, the function prioritizes accordingly:
    1. start of the month/year
    2. end of the month/year
    3. middle of the month/year
- The imputation indicator variable(s) for the imputation level is already defined in the data set and initialized to null. *DTF represents the date imputation indicator for the corresponding *DT variable which is based on the SDTM --DTC variable. *DTF is set to the highest level of missingness:
    - Y if the year is imputed
    - M if the month is imputed
    - D if the day is imputed
    - Null if there is no imputation

BUILDING CUSTOM FUNCTION

The user-defined function requires four arguments:

- **DATTIM** represents the SDTM datetime variable (i.e., --DTC, --STDTC, --ENDTC)
- **REFDT** represents the reference date (e.g., treatment start date, TRTSDT)
- **IMPUTFL** represents the imputation variable that will store the imputation level (i.e., ADTF, ASTDTF, AENDTF)
- **IMPTYP** represents the various imputation rules as seen in Table 1 and must be enclosed with quotations

CODE	IMPUTATION RULE
S	Start of month if day is missing Start of year if month is missing, regardless of whether day is missing or not
F	Start of month if day is missing Start of year if month is missing, regardless of whether day is missing or not
B	Start of month if day is missing Start of year if month is missing, regardless of whether day is missing or not
L	End of month if day is missing End of year if month is missing, regardless of whether day is missing or not
E	End of month if day is missing End of year if month is missing, regardless of whether day is missing or not
M	Middle of month if day is missing Middle of year if month is missing, regardless of whether day is missing or not
H	Middle of month if day is missing Middle of year if month is missing, regardless of whether day is missing or not
R	Impute to reference date if month/year matches reference – year must be non-missing. If imputation at the day level and the month and year are the same as the month and year of the reference date, then the analysis date is imputed to the reference date. If imputation at the month level and the year is the same as the year of reference date, then the analysis date is imputed to the reference date. If the year is missing, then no imputation is done.
Y	Same as 'R' but imputes to the reference date when year is missing. If imputation at the day level and the month and year are the same as the month and year of the reference date, then the analysis date is imputed to the reference date. If imputation at the month level and the year is the same as the year of reference date, then the analysis date is imputed to the reference date. If the year is missing, then then analysis date is imputed to the reference date.

Table 1: Arguments for Various Imputation Rules

In SDTM, the ISO 8601 date format is used (YYYY-MM-DD) with each missing component denoted using a single hyphen. The user-defined function extracts only the date portion of the DATTIM argument, ignoring any time component. Once the date is extracted, it is parsed into the three date components (i.e.,

year, month, and day).

The function then determines which imputation rules should be implemented based on the values included in the IMPTYP argument. Multiple imputation rules can be specified with multiple codes allowed for some rules. For example,

- To impute to the start of the month/year, use 'S', 'F', or 'B'
- To impute to the end of the month/year, use 'L' or 'E'
- To impute to the middle of the month/year, use 'M' or 'H'
- To use a reference date, include 'R' or 'Y'. Refer to Table 1 for details on difference between these two codes

Every blind date needs to have some guidelines; thus, if the date imputation is based on the start of the month/year as well as using the reference date, then any of the combinations can be used: 'SR', 'SY', 'YF', 'YB', 'RB', 'FR'. The order of the code used to indicate the imputation rule is not important nor are the code elements case sensitive.

After the imputation rules are determined, the function applies the imputation rule with the highest level of missingness and updates the IMPUTFL argument accordingly. The numeric date is returned using the **RETURN** statement and updates IMPUTFL via the **OUTARGS** statement. Refer to SAS Program 2 for illustration of the RETURN statement and OUTARGS. For the full code refer to the Appendix A.

```
proc fcmp outlib = fcmp.funcs.ISO_impdate;
   function isoimpdt(dattim $, refdt, imputfl $, imptyp $);
      outargs imputfl;

      length impdt 8 ___dt $10 __dtyr __dtmo __dtdy __impmo __impdy
             __impmos __impdys __impmoe __impdye __impmom __impdym 8
             imputfl __start __end __mid $1 __tempvar $2;
      format impdt date9.;

      /* VARIOUS SAS STATEMENTS TO PARSE DATE AND SET IMPUTATION RULES */

      if not missing(__dtyr) then do;
         if nmiss(__dtmo, __dtdy) = 0 then impdt = mdy(__dtmo, __dtdy, __dtyr);
         else if missing(__dtmo) then do;
            imputfl = 'M';
            /* IMPUTING DATE BASE ON IMPUTATION RULES */
         end;
         else if missing(__dtdy) then do;
            imputfl = 'D';
            /* IMPUTING DATE BASE ON IMPUTATION RULES */
         end;
      end;
      else if missing(__dtyr) then do;
         if find(imptyp, 'Y', 'i') then do;
            impdt = refdt;
            imputfl = 'Y';
         end;
         else call missing(impdt, imputfl);
      end;

      return(impdt);
   endfunc;
quit;
```

SAS Program 2: Portion of User-Defined Function ISOIMPDT

USING CUSTOM FUNCTIONS

Once the user-defined function has been defined, compiled, and saved for future use, the function is almost ready to be called up for your dates. To access the function, the **CMPLIB = LIBRARY** (where LIBRARY represents the previously compiled libraries, i.e., **LIBNAME.DATASET**, that are to be linked to in the program) needs to be specified on the system OPTION statement. The two-level name for the library specified in **CMPLIB=** must match the first two levels of the **OUTLIB=** in the PROC FCMP statement.

CMPLIB is akin to the LIBNAME statement. The LIBNAME statement associates a SAS library with a library-reference, which points to one or more locations where files are stored. CMPLIB is a reference to a location (or locations) where compiled functions and subroutines are stored. Essentially, CMPLIB tells SAS where the user-defined subroutines and functions are stored so that the functions and subroutines can be used like any SAS-defined function or subroutine.

Once you have specified this option, then the function can be used like any other SAS function as shown in SAS Program 3.

In this example, the start date (AESTDTC) is to be imputed based on the following combination of rules ("SEY"):

- **S -** If the day is missing, and month and year are non-missing, the date is imputed to the first of the month; if the month is missing and the year is non-missing, the date is imputed to the first of the year, i.e., January 1st.
- **E** - If the day is missing, the date is imputed to the end of the month; if the month is missing, the date is imputed to the end of the year, i.e., December 31st.
- **Y** - If the day is missing, and the month and year are the same as the month and year of the reference date, the date is imputed to the same value as the reference date; if the month is missing and the year is the same as the year of the reference date, the date is imputed to the same value as the reference date; lastly, if year is missing, the date is imputed to the same value as the reference date.

ASTDTF is the variable that represents the level of missingness and is updated when the function is executed.

For the end date (AEENDTC), a similar approach is taken based on the following combination of rules ("SE"):

- **S** - If the day is missing, and month and year are non-missing, the date is imputed to the first of the month; if the month is missing and year is non-missing, the date is imputed first of the year.
- **E** - If the day is missing, the date is imputed to the end of the month; and if month is missing, the date is imputed to the end of the year.

When the start of the month/year and the end of the month/year are both specified as rules, the partial date is imputed as follows:

- If the partial date is determined to be on or after the reference date, then the start of the month/year takes precedence over the end of the month/year.
- If the partial date is determined to be prior to the reference date, then the end of the month/year is used to impute the date.

The results from SAS Program 3 are shown in Data Display 2. For more examples, refer to Appendix B.

```
options cmplib = fcmp.funcs;
data impdt;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'sey');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'se');
run;
```

SAS Program 3: Using ISOIMPDT to Convert Dates to Numeric Values and Set Imputation Level

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	31DEC2021	31DEC2021	M	M
2021-02	2021-03	02JAN2022	28FEB2021	31MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	01SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	31JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	31DEC2021	01JAN2022	M	M
2021-07--T12	2022	02JAN2022	31JUL2021	01JAN2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	01AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	02JAN2022	01FEB2022	M	D
2022-01	2024-02	02JAN2022	02JAN2022	01FEB2024	D	D
	2022	02JAN2022	02JAN2022	01JAN2022	Y	M
2021---09T-:15	2022-08-09	02JAN2022	31DEC2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	02JAN2022	.	Y	
-----T12:15	-----T12:15	02JAN2022	02JAN2022	.	Y	
----09	----09	02JAN2022	02JAN2022	.	Y	
2023	2023	02JAN2022	01JAN2023	01JAN2023	M	M
2023-02	2023-03	02JAN2022	01FEB2023	01MAR2023	D	D

Data Display 2: Numeric Conversion and Imputation

Note that the function does not do a cross-check against multiple dates being imputed. If your dealing with multiple dates and there is dependency, then special care needs to be taken to ensure that the dates are in the correct order. For example, if your data has a start date and an end date that may require imputation, then additional handling rules need to be applied so that the start date comes before the end date. Afterall, you don't want one date taking place before the other has ended.

CONCLUSION

Even with the numerous SAS functions and subroutines available, there are situations where a specific need requires something custom. This is where the power of PROC FCMP comes into play: programmers can build their own functions and subroutines and use them just like SAS-defined functions and subroutines. This paper demonstrates how programmers can use PROC FCMP to ensure that incomplete dates don't leave their data sets heartbroken. Whether it's filling in the gaps with a first-of-the-month fix, rekindling a connection with a reference date, or committing to a full year when only part of the date shows up, these imputation rules keep things consistent. Just like a great relationship, a well-defined function is reliable, adaptable, and built to last ensuring you get that second date.

REFERENCES AND RECOMMENDED READINGS

Carpenter, A. (2013). Using PROC FCMP to the Fullest: Getting Started and Doing More. *SAS Global Forum.* San Francisco. Retrieved Apr 2022, from https://support.sas.com/resources/papers/proceedings13/139-2013.pdf

Eberhardt, P. (2011). A Cup of Coffee and Proc FCMP: I Cannot Function Without Them. *PharmaSUG.* Nashville: PharmaSUG. Retrieved Apr 2022, from https://www.lexjansen.com/pharmasug/2011/TU/PharmaSUG-2011-TU07.pdf

Hughes, T. M. (2022). *SAS® Data-Driven Development: From Abstract Design to Dynamic Functionality, Second Edition.* Troy Martin Hughes.

Hughes, T. M. (2024). *PROC FCMP User-Defined Functions: An Introduction to the SAS® Function Compiler.* Cary, NC: SAS Institute, Inc.

ISO 8601 Date and Time Format. (n.d.). Retrieved Apr 2022, from International Organization for Standardization: https://www.iso.org/iso-8601-date-and-time-format.html

Morgan, D. (2017). ISO 8601 and SAS®: A Practical Approach. *MWSUG.* St. Louis: MWSUG. Retrieved Jun 2022, from https://www.mwsug.org/proceedings/2017/PH/MWSUG-2017-PH02.pdf

Morgan, D. P. (2014). *The Essential Guide to SAS® Dates and Times; Second Edition.* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (2019, 11 Apr). *About SAS Date, Time, and Datetime Values*. Retrieved Jun 2022, from SAS® 9.4 and SAS® Viya® 3.5 Programming Documentation: https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/lrcon/p1wj0wt2ebe2a0n1lv4lem9hdc0v.htm

SAS Institute Inc. (2022, Apr 14). *CALL IS8601_CONVERT Routine*. Retrieved Jun 2022, from SAS® 9.4 and SAS® Viya® 3.2 Programming Documentation: https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.2/lefunctionsref/p0bhy7ndmdivmmn10b2okmbgiqmj.htm#p0h41lprw0az65n1tafhcnsqndru

SAS Institute Inc. (2022, May 24). *FCMP Procedure*. Retrieved Jun 2022, from SAS® 9.4 and SAS® Viya® 3.2 Programming Documentation: https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/proc/p10b4qouzgi6sqn154ipglazix2q.htm

SAS Institute Inc. (2022, May 13). *Working with Dates and Times by Using the ISO 8601 Basic and Extended Notations*. Retrieved Jun 2022, from SAS® 9.4 and SAS® Viya® 3.5 Programming Documentation: https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/leforinforref/p1a0qt18rxydrkn1b0rtdfh2t8zs.htm

Watson, R. J. (2022). *Have a Date with ISO®? Using PROC FCMP to Convert Dates to ISO 8601*. Retrieved from https://tp1210.p3cdn1.secureserver.net/wp-content/uploads/2023/07/ISO8601_FCMP.pdf

Wilson, K. (2012). Harnessing the Power of SAS ISO 8601 Informats, Formats, and the CALL IS8601_CONVERT Routine. *PhramaSUG.* San Francisco: PharmaSUG. Retrieved Jun 2022, from https://www.lexjansen.com/pharmasug/2012/DS/PharmaSUG-2012-DS22-SAS.pdf

ACKNOWLEDGMENTS

Thanks to Inka Leprince for testing the function.

Thanks to Inka Leprince, Louise Hadden, and Troy Martin Hughes for helping me come up with ‘perfect’ way to phrase my thoughts so that I was able to adequately convey a concept.

Thanks to Troy Martin Hughes for being so patient with me as I sent him copious text messages asking for his advice and input and for reviewing this paper to ensure that I was not misstating anything.

CONTACT INFORMATION

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

Richann Jean Watson
DataRich Consulting
richann.watson@datarichconsulting.com

APPENDIX A: USER-DEFINED FUNCTIONS FOR CONVERTING AND IMPUTING ISO 8601 DATES TO NUMERIC DATES

```
proc fcmp outlib = fcmp.funcs.ISO_impdate;
   function isoimpdt(dattim $, refdt, imputfl $, imptyp $);
      outargs imputfl;
      length impdt 8 ___dt $10 __dtyr __dtmo __dtdy __impmo __impdy
             __impmos __impdys __impmoe __impdye __impmom __impdym 8
             imputfl __start __end __mid $1 __tempvar $2;
      format impdt date9.;

      /* extract the date portion only */
      ___dt = strip(scan(dattim, 1, 'T'));

      /* if year is missing then impute the entire date */
      if ___dt =: '-' then call missing(of __dt:);
      else if anydigit(first(___dt)) then do;
         __dtyr = input(substr(___dt, 1, 4), best.);
         if substr(___dt, 6, 1) = '-' then call missing(__dtmo, __dtdy);
         else if anydigit(substr(___dt, 6, 1)) then do;
            __dtmo = input(substr(___dt, 6, 2), best.);
            if anydigit(substr(___dt, 9, 1)) then
               __dtdy = input(substr(___dt, 9, 2), best.);
            else call missing(__dtdy);
         end;
      end;

      /* determine the imputation month and imputation day number */
      if prxmatch('/S|F|B/i', imptyp) then do;
         __start = 1;
         __impmos = 1;
         __impdys = 1;
      end;
      if prxmatch('/L|E/i', imptyp) then do;
         __end = 1;
         __impmoe = 12;
         if not missing(__dtmo) and __dtmo ne 12
              then __impdye = day(mdy(__dtmo + 1, 1, __dtyr) - 1);
         else if missing(__dtmo) or __dtmo = 12 then __impdye = 31;
      end;
      if prxmatch('/M|H/i', imptyp) then do;
         __mid = 1;
         __impmom = 6;
         __impdym = 15;
      end;
      __impmo = coalesce(__impmos, __impmoe, __impmom);
      __impdy = coalesce(__impdys, __impdye, __impdym);

      /* impute dates based on the following rules: denoted in IMPTYP argument */
      if not missing(__dtyr) then do;
         if nmiss(__dtmo, __dtdy) = 0 then impdt = mdy(__dtmo, __dtdy, __dtyr);
         else if missing(__dtmo) then do;
            imputfl = 'M';
            if prxmatch('/R|Y/i', imptyp) and __dtyr = year(refdt) then impdt = refdt;
            else if __dtyr < year(refdt) and __end = 1 then
                    impdt = mdy(__impmoe, __impdye, __dtyr);
            else if __dtyr < year(refdt) and __mid = 1 then
                    impdt = mdy(__impmom, __impdym, __dtyr);
            else impdt = mdy(__impmo, __impdy, __dtyr);
         end;
         else if missing(__dtdy) then do;
             imputfl = 'D';
            if prxmatch('/R|Y/i', imptyp) and __dtyr = year(refdt) and
               __dtmo = month(refdt) then impdt = refdt;
            else if mdy(1, __dtmo, __dtyr) < mdy(1, month(refdt), year(refdt)) and
                    __end = 1 then impdt = mdy(__dtmo, __impdye, __dtyr);
            else if mdy(1, __dtmo, __dtyr) < mdy(1, month(refdt), year(refdt)) and
```

```
                    __mid = 1 then impdt = mdy(__dtmo, __impdym, __dtyr);
               else impdt = mdy(__dtmo, __impdy, __dtyr);
            end;
         end;
         else if missing(__dtyr) then do;
            if find(imptyp, 'Y', 'i') then do;
               impdt = refdt;
               imputfl = 'Y';
            end;
            else call missing(impdt, imputfl);
         end;
         return(impdt);
      endfunc;
quit;
```

APPENDIX B: EXAMPLES

Below are several examples that illustrate the use of the ISOIMPDT user-defined function along with the output using the data from Data Display 1.

In SAS Program 4, both the start and end dates are imputed to the middle of the month (for missing day) and to the middle of the year (for missing month). However, with the end date, if the month and day are the same as the month and day of the reference date then it is imputed to the reference date; and if the month is missing and the year is the same as the year of the reference date the date is imputed to the reference date. Data Display 3 shows the results for SAS Program 4. Notice the use of 'M' and 'H'. Both represent the middle of the month/year.

```
data impdt;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'M');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'HR');
run;
```

SAS Program 4: Impute Start and End to Middle of Month/Year with End Using the Reference Date

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	15JUN2021	15JUN2021	M	M
2021-02	2021-03	02JAN2022	15FEB2021	15MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	15SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	15JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	15JUN2021	02JAN2022	M	M
2021-07--T12	2022	02JAN2022	15JUL2021	02JAN2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	15AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	15JUN2022	15FEB2022	M	D
2022-01	2024-02	02JAN2022	15JAN2022	15FEB2024	D	D
	2022	02JAN2022	.	02JAN2022		M
2021---09T-:15	2022-08-09	02JAN2022	15JUN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	.	.		
-----T12:15	-----T12:15	02JAN2022	.	.		
----09	----09	02JAN2022	.	.		
2023	2023	02JAN2022	15JUN2023	15JUN2023	M	M
2023-02	2023-03	02JAN2022	15FEB2023	15MAR2023	D	D

Data Display 3: Output for SAS Program 4

SAS Program 5 is similar to SAS Program 4 in that both the start and end dates are imputed to the middle of the month (for missing day) and to the middle of the year (for missing month). However, the start date uses the reference date when the month and day are the same as the month and day of the reference date, and when the month is missing and the year is the same as the year of the reference date. Data Display 4 shows the results for SAS Program 5.

```
data impdt2;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'MR');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'H');
run;
```

SAS Program 5: Impute Start and End to Middle of Month/Year with Start Using the Reference Date

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	15JUN2021	15JUN2021	M	M
2021-02	2021-03	02JAN2022	15FEB2021	15MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	15SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	15JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	15JUN2021	15JUN2022	M	M
2021-07--T12	2022	02JAN2022	15JUL2021	15JUN2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	15AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	02JAN2022	15FEB2022	M	D
2022-01	2024-02	02JAN2022	02JAN2022	15FEB2024	D	D
	2022	02JAN2022	.	15JUN2022		M
2021---09T-:15	2022-08-09	02JAN2022	15JUN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	.	.		
-----T12:15	-----T12:15	02JAN2022	.	.		
----09	----09	02JAN2022	.	.		
2023	2023	02JAN2022	15JUN2023	15JUN2023	M	M
2023-02	2023-03	02JAN2022	15FEB2023	15MAR2023	D	D

Data Display 4: Output for SAS Program 5

SAS Program 6 yields similar results as SAS Program 5 with the difference being that SAS Program 6 uses the reference date to populate missing dates, as seen in Data Display 5.

```
data impdt3;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'MY');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'M');
run;
```

SAS Program 6: Impute Start and End to Middle of Month/Year with Start Using Reference Date for Missing Dates

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	15JUN2021	15JUN2021	M	M
2021-02	2021-03	02JAN2022	15FEB2021	15MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	15SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	15JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	15JUN2021	15JUN2022	M	M
2021-07--T12	2022	02JAN2022	15JUL2021	15JUN2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	15AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	02JAN2022	15FEB2022	M	D
2022-01	2024-02	02JAN2022	02JAN2022	15FEB2024	D	D
	2022	02JAN2022	02JAN2022	15JUN2022	Y	M
2021---09T-:15	2022-08-09	02JAN2022	15JUN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	02JAN2022	.	Y	
-----T12:15	-----T12:15	02JAN2022	02JAN2022	.	Y	
----09	----09	02JAN2022	02JAN2022	.	Y	
2023	2023	02JAN2022	15JUN2023	15JUN2023	M	M
2023-02	2023-03	02JAN2022	15FEB2023	15MAR2023	D	D

Data Display 5: Output for SAS Program 6

SAS Program 7 imputes the start to the beginning of the month/year using the imputation code of 'B' and the end date to the end of the month or year using the imputation code of 'E'. Neither use the reference date (Data Display 6).

```
data impdt4;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'b');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'e');
run;
```

SAS Program 7: Impute Start to the Beginning of the Month/Year and End to the End of the Month/Year

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	01JAN2021	31DEC2021	M	M
2021-02	2021-03	02JAN2022	01FEB2021	31MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	30SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	01JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	01JAN2021	31DEC2022	M	M
2021-07--T12	2022	02JAN2022	01JUL2021	31DEC2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	31AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	01JAN2022	28FEB2022	M	D
2022-01	2024-02	02JAN2022	01JAN2022	29FEB2024	D	D
	2022	02JAN2022	.	31DEC2022		M
2021---09T-:15	2022-08-09	02JAN2022	01JAN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	.	.		
-----T12:15	-----T12:15	02JAN2022	.	.		
----09	----09	02JAN2022	.	.		
2023	2023	02JAN2022	01JAN2023	31DEC2023	M	M
2023-02	2023-03	02JAN2022	01FEB2023	31MAR2023	D	D

Data Display 6: Output for SAS Program 7

Like SAS Program 7, SAS Program 8 imputes the start date to the beginning of the month/year but uses the imputation code 'F' instead of 'B' but also uses the reference date if the month/year is the same as the month/year of the reference date or if the month is missing and the year is the same as the year of the reference date. The end date is imputed to the end of the month/year but uses the imputation code of 'L'. Results are shown in Data Display 7.

```
data impdt5;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'fr');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'l');
run;
```

SAS Program 8: Impute Start to the Beginning of the Month/Year and Uses the Reference Date and End to the End of the Month/Year

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	01JAN2021	31DEC2021	M	M
2021-02	2021-03	02JAN2022	01FEB2021	31MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	30SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	01JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	01JAN2021	31DEC2022	M	M
2021-07--T12	2022	02JAN2022	01JUL2021	31DEC2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	31AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	02JAN2022	28FEB2022	M	D
2022-01	2024-02	02JAN2022	02JAN2022	29FEB2024	D	D
	2022	02JAN2022	.	31DEC2022		M
2021---09T-:15	2022-08-09	02JAN2022	01JAN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	.	.		
-----T12:15	-----T12:15	02JAN2022	.	.		
----09	----09	02JAN2022	.	.		
2023	2023	02JAN2022	01JAN2023	31DEC2023	M	M
2023-02	2023-03	02JAN2022	01FEB2023	31MAR2023	D	D

Data Display 7: Output for SAS Program 8

SAS Program 9 imputes the start date to the beginning of the month/year and the end date to the end of the month/year, and also uses the reference date to populate missing start dates. Refer to Data Display 8 for results.

```
data impdt6;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'yf');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'l');
run;
```

SAS Program 9: Impute Start to the Beginning of the Month/Year and Uses the Reference Date for Missing Dates and End to the End of the Month/Year

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	01JAN2021	31DEC2021	M	M
2021-02	2021-03	02JAN2022	01FEB2021	31MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	30SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	01JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	01JAN2021	31DEC2022	M	M
2021-07--T12	2022	02JAN2022	01JUL2021	31DEC2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	31AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	02JAN2022	28FEB2022	M	D
2022-01	2024-02	02JAN2022	02JAN2022	29FEB2024	D	D
	2022	02JAN2022	02JAN2022	31DEC2022	Y	M
2021---09T-:15	2022-08-09	02JAN2022	01JAN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	02JAN2022	.	Y	
-----T12:15	-----T12:15	02JAN2022	02JAN2022	.	Y	
----09	----09	02JAN2022	02JAN2022	.	Y	
2023	2023	02JAN2022	01JAN2023	31DEC2023	M	M
2023-02	2023-03	02JAN2022	01FEB2023	31MAR2023	D	D

Data Display 8: Output for SAS Program 9

SAS Program 10 imputes the start to the beginning of the month/year using the imputation code ‘S’ and the end to the middle of the month/year and neither use the reference date (Data Display 9).

```
data impdt7;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 's');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'h');
run;
```

SAS Program 10: Impute Start to the Beginning of the Month/Year and End to the Middle of the Month/Year

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	01JAN2021	15JUN2021	M	M
2021-02	2021-03	02JAN2022	01FEB2021	15MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	15SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	01JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	01JAN2021	15JUN2022	M	M
2021-07--T12	2022	02JAN2022	01JUL2021	15JUN2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	15AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	01JAN2022	15FEB2022	M	D
2022-01	2024-02	02JAN2022	01JAN2022	15FEB2024	D	D
	2022	02JAN2022	.	15JUN2022		M
2021---09T-:15	2022-08-09	02JAN2022	01JAN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	.	.		
-----T12:15	-----T12:15	02JAN2022	.	.		
----09	----09	02JAN2022	.	.		
2023	2023	02JAN2022	01JAN2023	15JUN2023	M	M
2023-02	2023-03	02JAN2022	01FEB2023	15MAR2023	D	D

Data Display 9: Output for SAS Program 10

SAS Program 11 illustrates the use of two imputation codes for start dates. If the month/year is prior to the month/year of the reference date or if the month is missing and the year is prior to the year of the reference date, then the date is imputed to the middle of the month/year. If the month/year is the same as the month/year of the reference or the if the month is missing and the year is the same as the year of the reference date, then the date is imputed to the reference date. If the date is completely missing, then the date is imputed to the reference date. Otherwise, the date is imputed to the start of the month/year. For the end date, it is imputed to the end of the month/year and uses the reference date. Results can be seen in Data Display 10.

```
data impdt8;
   set dates;
   format ASTDT AENDT date9.;
   length ASTDTF AENDTF $1;
   call missing(ASTDTF, AENDTF);

   ASTDT = isoimpdt(AESTDTC, TRTSDT, ASTDTF, 'smy');
   AENDT = isoimpdt(AEENDTC, TRTSDT, AENDTF, 'er');
run;
```

SAS Program 11: Impute Start to the Beginning of the Month/Year if Partial Date is On or After Reference Date and Middle of the Month/Year if Prior to Reference Date and Uses Reference Date for Missing Dates and Imputes End to End of Month/Year and Uses Reference Date

AESTDTC	AEENDTC	TRTSDT	ASTDT	AENDT	ASTDTF	AENDTF
2021	2021	02JAN2022	15JUN2021	31DEC2021	M	M
2021-02	2021-03	02JAN2022	15FEB2021	31MAR2021	D	D
2021-07-09	2022-09	02JAN2022	09JUL2021	30SEP2022		D
2021-07-09T12:15	2021-09-09T12:15	02JAN2022	09JUL2021	09SEP2021		
2021-07	2022-08-09	02JAN2022	15JUL2021	09AUG2022	D	
2021	2022---09	02JAN2022	15JUN2021	02JAN2022	M	M
2021-07--T12	2022	02JAN2022	15JUL2021	02JAN2022	D	M
2021-07-09T12:15		02JAN2022	09JUL2021	.		
2021-07-09T12:15:45	2022-08--T12:15	02JAN2022	09JUL2021	31AUG2022		D
2021-07-09T-:15	2022-08-09	02JAN2022	09JUL2021	09AUG2022		
2022	2022-02	02JAN2022	02JAN2022	28FEB2022	M	D
2022-01	2024-02	02JAN2022	02JAN2022	29FEB2024	D	D
	2022	02JAN2022	02JAN2022	02JAN2022	Y	M
2021---09T-:15	2022-08-09	02JAN2022	15JUN2021	09AUG2022	M	
--07-09	--08-09	02JAN2022	02JAN2022	.	Y	
-----T12:15	-----T12:15	02JAN2022	02JAN2022	.	Y	
----09	----09	02JAN2022	02JAN2022	.	Y	
2023	2023	02JAN2022	01JAN2023	31DEC2023	M	M
2023-02	2023-03	02JAN2022	01FEB2023	31MAR2023	D	D

Data Display 10: Output for SAS Program 11